



VYSOKÁ ŠKOLA BÁŇSKÁ – TECHNICKÁ UNIVERZITA OSRAVA

FAKULTA ELEKTROTECHNIKY A INFORMATIKY

KATEDRA KYBERNETIKY A BIOMEDICÍNSKÉHO INŽENÝRSTVÍ

Měření a zpracování dechové křivky v plánování radioterapie

Measurement and Processing of Breath Signal for Planning Radiotherapy

Student: Markéta Nováková

Vedoucí bakalářské práce: Ing. Marek Penhaker, Ph.D.

Ostrava 2015

## Zadání bakalářské práce

Student: **Markéta Nováková**

Studijní program: B2649 Elektrotechnika

Studijní obor: 3901R039 Biomedicínský technik

Téma: **Měření a zpracování dechové křivky v plánování radioterapie**  
**Measurement and Processing of Breath Signal for Planning**  
**Radiotherapy**

### Zásady pro vypracování:

1. Seznámení se s problematikou radioterapie.
2. Seznámení se s problematikou robotického radiochirurgického systému CyberKnife a možností plánování léčby.
3. Návrh a realizace postupů pro extrakci reálné dechové křivky pacienta.
4. Stanovení vhodné kombinace parametrů plánování radioterapie pro pohyblivá ložiska.
5. Provedení měření a testů.
6. Vyhodnocení výsledků.

### Seznam doporučené odborné literatury:

- [1] SVATOŠ, Josef. *Biologické signály I*. Praha: ČVUT Praha, 1995. 204 s. ISBN 8001008843.
- [2] PENHAKER, M., M. IMRAMOVSKÝ, P. TIEFENBACH a F. KOBZA. *Lékařské diagnostické přístroje učební texty*. Ostrava: VŠB - TU Ostrava, 2004. 320 s. ISBN 80-248-0751-3.
- [3] PEREZ, Reinaldo. *Design of Medical Electronic Devices*. 1st Edition. San Diego(USA): Academic Press, 2002. 296 s. ISBN 0125507119.
- [4] WEBSTER, John G., ed. *Medical instrumentation: Application and Design*. 3rd Edition. NY(USA): John Wiley&Sons, 1997. 691 s. ISBN 0471153680.
- [5] CARR, Joseph J. and John M. BROWN. *Introduction to Biomedical Equipment Technology*. 4th Edition. Upper Saddle River, New Jersey(USA): Prentice Hall, 2000. 743 s. ISBN 0130104922.
- [6] MACKAY, R. Stuart. *Bio-Medical Telemetry: Sensing and Transmitting Biological Information from Animals and Man*. 2nd ed., Revised. Hoboken: John Wiley, 1998. 556 s. ISBN 0780347188.
- [7] FRADEN, Jacob. *Handbook of Modern Sensors: Physics, Designs, and Applications (Handbook of Modern Sensors)*. 3rd Edition. NY(USA): Springer, 2003. 589 s. ISBN 0387007504.
- [8] OLANSEN, Jon B. and Eric ROSOW. *Virtual Bio-Instrumentation: Biomedical, Clinical, and Healthcare Applications in LabVIEW*. Upper Saddle River(USA): Prentice Hall, 2002. 608 s. ISBN 0130652164.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Marek Penhaker, Ph.D.**

Datum zadání: 01.09.2013

Datum odevzdání: 07.05.2015



---

doc. Ing. Jiří Koziorek, Ph.D.  
*vedoucí katedry*



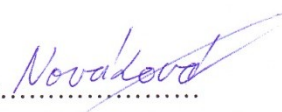
---

prof. RNDr. Václav Snášel, CSc.  
*děkan fakulty*

Prohlašuji, že jsem celou práci, včetně všech příloh, vypracovala samostatně.

Uvedla jsem všechny literární prameny a publikace, ze kterých jsem čerpala.

Ostrava 6.5.2015.....

Podpis: .....

## **Poděkování**

Tímto bych chtěla poděkovat vedoucímu mé bakalářské práce panu **Ing. Mareku Penhakerovi, Ph.D.** za odbornou pomoc a konzultace při vytváření práce. Zároveň bych chtěla poděkovat panu **Ing. Lukášovi Knybelovi** za poskytnutá data a konzultace, kterými přispěl k vytvoření této práce.

## **Abstrakt**

Cílem této práce je vznik softwaru v programu Matlab, který bude v grafickém rozhraní (GUI) zobrazovat načtení souborů vznikajících při ozařování radiologickým přístrojem CyberKnife. Tato data by měl program dokázat vhodně upravit a vyčíst z nich důležité informace.

Program by měl přispět k ověření správnosti nastavení ozařovaných lemů nádoru, které jsou používány v konvenční radioterapii, protože bude díky němu možné zpracovat data přístroje CyberKnife, která zjišťují pohyby pacienta při dýchání v průběhu léčby. Přispět by měl ke zkvalitnění léčby ozařovaných nádorů díky následnému možnému statistickému zpracování dat.

## **Klíčová slova**

Radioterapie, dechová křivka, plánování radioterapie, respirace v radioterapii, CyberKnife, nežádoucí pohyby v radioterapii

## **Abstract**

Aim of this work is the developing of software in Matlab, which will display retrieving of files in a graphical interface (GUI). Files are generated during radiation with radiological device CyberKnife. The program should prove and conveniently edit this data and read all important information.

The program should help to verify the accuracy of the settings radiated tumor rims, which are used in conventional radiotherapy. Thanks to it, it is possible to process data of CyberKnife device which detect movements of patient while breathing. Program should contribute to improving of the treatment of irradiated tumors with possible subsequent statistical evaluation.

## **Keywords**

Radiotherapy, Breath Signal, Radiotherapy Signal, Respiration in Radiotherapy, CyberKnife, Unfavorable Movements in Radiotherapy

## Seznam použitých symbolů a zkratek

RTG	Rentgenové záření
MeV	Megaelectron-volt
CT	Počítačová tomografie
2D	Dvourozměrný
3D	Trojrozměrný
DNA	Deoxyribonukleová kyselina
RNA	Ribonukleová kyselina
Log	Typ souboru pořizovaný přístrojem CyberKnife
FIR	Filtr s konečnou impulzní odezvou
IIR	Filtr s nekonečnou impulzní odezvou
NaN	Bezčíselná hodnota (Not-a-Number)
GUI	Grafické uživatelské rozhraní (Graphical User Interface)



## Seznam Obrázků

- Obrázek 1 - Plicní objemy
- Obrázek 2 - CyberKnife
- Obrázek 3 - Plánovací systém přístroje CyberKnife u plicního případu
- Obrázek 4 - Původní soubor typu log
- Obrázek 5 - Osa x v závislosti na čase
- Obrázek 6 - Upravená data bez závadných znaků.
- Obrázek 7 - Tabulka s datovým souborem, na který byla použita funkce textscen
- Obrázek 8 - Rotace osy x a y o 45°
- Obrázek 9 - Popis os vůči robotovi a vůči pacientovi
- Obrázek 10 - Výstupní data po kompletní úpravě
- Obrázek 11 - Uložená data v pořadí čas, osa x, osa y, osa z a interval.
- Obrázek 12 - Řádek a hodnota přerušení léčby jak ji uloží program
- Obrázek 13 - Data vyčtená ze souboru stručně uložená do struktury.
- Obrázek 14 - Načtená přibližná křivka z původních dat
- Obrázek 15 - Použití FIR2 filtru s nastavením vektoru  $f=[0 \ 0.5 \ 0.5 \ 1]$
- Obrázek 16 - Použití FIR2 filtru s nastavením  $f=[0 \ 0.15 \ 0.15 \ 1]$
- Obrázek 17 - Hledání extrémů, hodnotu si zachová pouze extrém
- Obrázek 18 - Načtená dechová křivka bez funkce MaxMin
- Obrázek 19 - Použití funkce MaxMin s nastaveným rozdílem mezi dvěma extrémy 30
- Obrázek 20 - Ukázka načteného souboru v programu
- Obrázek 21 - Chyba v extrému
- Obrázek 22 - Ukázka načteného souboru v programu
- Obrázek 23 - Zobrazení extrému po stisknutí tlačítka „extrémy“
- Obrázek 24 - Umístění exportovaného souboru
- Obrázek 25 - Exportovaná tabulka zobrazená v programu Microsoft Excel
- Obrázek 26 - Chybová hláška v programu
- Obrázek 27 - Načtení souboru používaného od června roku 2014
- Obrázek 28 - Ukázka logu, nesouhlasné hodnoty časů
- Obrázek 29 - Rozdíly v čase

## Seznam tabulek

- Tabulka 1 - Hlavní studie hodnotící pohyby orgánů při volném dýchání
- Tabulka 2 - Testování funkčnosti programu

# Obsah

1	Úvod.....	2
2	Radioterapie .....	4
2.1	Plánování radioterapie.....	5
2.2	Brachyradioterapie .....	6
2.3	Teleterapie.....	6
2.4	Ozáření tkáně a jeho důsledky .....	6
3	Fyziologie dýchání.....	8
3.1	Nežádoucí vlivy dýchání a jejich eliminace pro radioterapii .....	9
4	Radiologický přístroj CyberKnife.....	12
4.1	Získávání dechové křivky .....	12
4.2	Léčba na přístroji CyberKnife.....	13
5	Zpracování dechové křivky.....	15
5.1	Instalace knihoven programu Matlab.....	16
5.2	Získání dat z datového souboru typu log z přístroje CyberKnife .....	16
5.3	Zjištění informací z hlavičky logu .....	17
5.4	Přepočítání úhlu křivky .....	18
5.5	Čas a počet přerušení .....	19
5.6	Struktura.....	21
5.6	Filtrování dat.....	22
5.7	Extrémy .....	24
5.8	Program v grafickém rozhraní GUI .....	28
5.9	Zobrazení extrémů .....	29
5.10	Export dat.....	30
5.11	Chybová hláška .....	32
5.12	Testování programu .....	32
6	Závěr .....	38

# 1 Úvod

V současnosti se velká část výzkumu medicíny zaměřuje na co nejefektivnější léčbu nádorů. Ve spojitosti s tím byl do provozu uveden ozařovací přístroj CyberKnife, který nyní řadíme k tomu nejlepšímu, co k ozařování nádorové tkáně bylo vyvinuto. Patří k nejpřesnější technice na desetinu centimetru. Avšak i přesto dochází k chybám a ke zbytečnému zasažení zdravé tkáně. A proto se stále hledají nové techniky, které by dokázaly těmto poškozením zabránit. Přístroj CyberKnife může také v mnohém přispět ve srovnání s léčbou, která byla užívána jako konvenční radioterapie a může nám ukázat, zda její nastavené parametry byly dostačující i při pohybu ložiska způsobeném dýcháním a dalšími nežádoucími vlivy. Tyto pohyby jsou fyziologickou součástí organismu, které nelze potlačit ani vyloučit.

Cílem této práce je přispět k poznatkům o pohybu nádorového ložiska v průběhu léčby. Za pomoci sledování pacientovy dechové křivky, která vzniká díky přístroji CyberKnife v průběhu ozařování pacienta, odhalit výchyly od klidového dýchání pacienta, aby nedocházelo ke zbytečnému ozáření zdravé tkáně a tato terapie tak byla mnohem přesnější. Přístroj CyberKnife dokáže nežádoucí pohyby sám kompenzovat na milimetrové úrovni. U konvenční radioterapie tomu tak ale není a k nádoru je přičítán velký bezpečnostní lem. Právě srovnání s daty vzniklých tímto programem by mělo být možné porovnat, zda jsou současně nastavené lemy u konvenční radioterapie správné, či v kolika procentech případu neodpovídají, kvůli fyziologickým pohybům v těle pacienta.

Soubory vytvořené přístrojem CyberKnife jsou opatřeny hlavičkou s údaji o léčbě a samotném pacientovi. Je nutné, aby program uměl tato data zpracovat a zobrazit. Pod hlavičkou jsou načítána data samotné dechové křivky, a to v trojrozměrném systému. Je tedy možné pozorovat pohyb ložiska v prostoru. V datech jsou znaky, které ale pro zpracování programem Matlab nejsou adekvátní. Je proto nutné upravit samotný datový soubor. Data jsou načítána v poloze vzhledem k robotovi přístroje CyberKnife, nikoli vůči pacientovi a je proto zapotřebí, aby program data před jejich načtením vyotoval.

Hlavním cílem této práce je vyvinutí softwaru, který by data dokázal upravit a zobrazit požadované informace, mezi které patří informace o léčbě, informace o pacientovi. Program by měl být schopen vyčistit data dechové křivky od překmitů použitím vhodného filtru, zobrazit amplitudy v dechové křivce a vyčíslit statistické údaje o přerušení léčby, době každého úseku a počet hlubokých nádechů a výdechů, které je nutno nastavit vhodnou funkcí. Důležitým údajem je také součet všech úseků, který vypovídá o době trvání celé léčby. Mezi další požadavky patří zobrazení pouhých amplitud křivky, tedy jejich extrémů,

které odpovídají maximu v nádechu a minimu ve výdechu. Posledním požadavkem na výstup softwaru je možnost exportování upravených hodnot do souboru, který by bylo možno nadále zpracovat matematickým softwarem.

V případě vytvoření tohoto softwaru, bude možno jej použít pro samotné plánování radioterapie u pacientů a celá léčba by se tak mohla stát opět o něco přesnější. Poslouží pro zpracování již proběhnuté léčby a statistickému zpracování pohybů pacienta a fyziologických pohybů v jeho těle v průběhu léčby, které ovlivňují samotný průběh ozařování. Bude sloužit pro kontrolu při konvenční radioterapii, aby bylo dokázáno, zda jsou současně nastavené hodnoty bezpečnostního lemu dostatečné. Veškerá data bude možno nadále zpracovávat a získávat lepší poznatky.

## 2 Radioterapie

Radioterapie je léčebná metoda využívající gama, RTG a elektronové záření, které spadá do skupiny elektromagnetických záření. Využívají se velká spektra záření o různé energii. Díky tomu je možné přesné nasměrování s cílem doručit co nejpresnější dávku k cílenému ložisku.<sup>[1]</sup> RTG záření má nízkou energii a většinu pohltí kůže, proto se nemůže využívat u nádorů, které jsou uloženy v hloubce. Řešení přišlo v 50. letech. Začaly se využívat převážně kobaltové ozařovače, které mají vysokou energii, a proto již mohou ozařovat hluboko uložené nádory. Současná radioterapie nejčastěji využívá lineární urychlovače, kde je možno nastavit množství energie a řídit tak celý proces ozařování s co možná nejmenšími postihy okolní tkáně.<sup>[2]</sup> Urychlovače dokážou urychlovat částice za pomoci vysokofrekvenčních elektromagnetických vln v lineární trubici.<sup>[3]</sup>

Energie záření se pohybuje v jednotkách, maximálně v desítkách MeV. Cílem radioterapie je účinné ozáření nádorového ložiska, aniž by došlo k poškození okolních tkání. Dosáhnout toho lze zejména dodržáním dvou faktorů. Prvním z nich je ozařování ložiska kolimovaným svazkem z více směrů. Ložisko zde představuje izocentrum z sečené dávky paprsků. Druhým způsobem je frakcionované ozařování, kdy se dávka rozdělí do většího počtu menších denních dávek. Poté dochází k ozařování po dobu několika týdnů. Toho lze využít díky poznatku, že zdravá tkáň rychleji regeneruje a nádorová tkáň regeneruje podstatně pomaleji a z důvodu svého dělení bývá citlivější na samotné ozáření.<sup>[4]</sup>

V současné době patří radioterapie mezi jednu z neúčinnějších léčebných metod nádorových onemocnění pro mnoho zjištěných a lokalizovaných nádorů. Mnohdy je právě radioterapie vhodná alternativa chirurgického výkonu.<sup>[5]</sup> Podmínkou pro ozáření je citlivost nádoru na samotné záření, což je zjistitelné pomocí histologického vyšetření po odebrání části tkáně. Posuzuje se poté zhoubnost nádoru, a zdali následná léčba bude prováděna radioterapií, či chemoterapií.<sup>[6]</sup>

Radioterapie může být využívána kdykoli během léčby, před operací, po ní, nebo jako samotný léčebný proces u nádorů, které již nelze operovat. Může se také využívat ve spojení s chemoterapií.<sup>[3]</sup>

Radioterapie bývá často využívána po samotném chirurgickém výkonu, kdy byla již všechna zjištěná ložiska odstraněna. Není nutná pouze u nádorů, které byly chirurgicky odstraněny včas. V případě použití následné léčby nádorového ložiska za pomoci radioterapie se jedná o adjuvantní radioterapii, což je pokračování léčby, která by měla

zabránit šíření nádorů ze zbytků ložiska, které mohly po operačním zákroku v těle pacienta zůstat. Také je cílená na metastázy daného ložiska.<sup>[8]</sup>

## 2.1 Plánování radioterapie

Před samotným plánováním radioterapie musí pacient podstoupit řadu klinických vyšetření, aby pro něj byla vybrána vhodná léčba.

Jelikož je zapotřebí ozářit nepravidelné trojrozměrné těleso, je nutná správná lokalizace a zobrazení ložiska. Je důležité správně diagnostikovat jeho přesnou velikost a tvar a typ. Je možno využít CT obrazů, které poté zpracovávají současné 3D systémy a plánují léčbu v prostoru. Výhodou 3D plánování oproti 2D plánování je možnost uspořádání polí ve více rovinách. Velikost a tvar svazku je možné nastavovat filtry, klíny a kolimátory. Poté je zapotřebí vyznačení cílových objemů a kritických orgánů, které by měly být ozářeny co nejméně, nebo vůbec. Zvolí se správná ozařovací technika a pomocí speciálních softwarů se spočítá dávka záření. Vše se nejdříve simuluje a optimalizuje. Aby nedošlo k dalšímu šíření nádoru, přidává se lem. To znamená, že se zvětší ozařované okolí ložiska, u konvenční léčby v řádech centimetrů, které by mohlo být zdrojem mikroskopických zbytků nádoru. Ty by mohly být zdrojem dalšího šíření.<sup>[3]</sup>

Důvodem, proč se u konvenční radioterapie nastavuje lem podstatně větší, jsou pohyby pacienta způsobené jeho dýcháním, nebo například fyziologickými pohyby orgánů při trávení, nebo také samotný pohybem pacienta. Jelikož není možné redukovat takto vzniklé pohyby, jsou brány v potaz už při samotném plánování. Musí se pokrýt celá plocha ložiska v nádechu i výdechu a proto jsou tyto hodnoty ve všech 3 směrech nádorového ložiska přiřítány. Způsobí to tak velkou plochu ozáření zdravé tkáně, kterou ale ve skutečnosti není nutné ozářit.<sup>[9]</sup>

Důležitý je způsob fixace pacienta, který je třeba zachovat při ozařování samotném, aby nedošlo změnou jeho polohy k jinému natočení ložiska a zbytečnému ozáření špatného objemu. Jelikož není možno pohybu pacienta úplně zabránit, přidává se kromě lemu ještě objem, který by měl kompenzovat fyziologické změny pozice pacienta a chyby při jeho nastavení.<sup>[3]</sup> Pacient bývá kvůli eliminaci pohybů fixován různými pomůckami, které jsou uzpůsobeny dle lokality, která je ozařována. U hlavy se jedná speciální masky, u nádorů prsů jsou to podložky k upevnění paže a hrudníku, nebo pěnové podložky u končetin.<sup>[10]</sup>

Každý ozařovací plán je optimalizován, aby byla dodána maximální dávka do ložiska s co nejmenším zasažením okolí léčeného objemu a co nejmenší dávkou pro celé tělo. Je zapotřebí zachovat Paddicův index, který je dán indexem a číslem konformity. K výpočtu se

používá plánovaný terčový objem, objem plánované izodózy a objem cíle, který je pokryt plánovanou izodózou. <sup>[11]</sup> Izodózy se využívají pro zobrazení dávky k cílenému ložisku a ozáření okolní tkáně, spojují se v místě se stejnou dávkou a v tomto místě je intenzita záření nejvyšší. V počítači pak dochází k zpracování těchto izodózních polí a vzniká ozařovací plán. <sup>[1]</sup>

## **2.2 Brachyradioterapie**

Tato metoda spočívá v přímém kontaktu zářiče a objemu. Je využívána pro ozáření menších objemů, jedná se tedy o lokální radioterapii. Zdroj záření se implantuje přímo do ložiska nádoru, na jeho povrch, nebo do tělesných dutin. Intenzita záření prudce klesá ve vzdálenosti od bezprostřední blízkosti záření. <sup>[4]</sup> Zavádí se na předem určenou dobu. Často bývá pokračováním léčby po chirurgickém výkonu, nebo teleterapii. <sup>[12]</sup>

## **2.3 Teleterapie**

Spadá do konzervativní léčby spolu s chemoterapií. Během teleterapie se využívá ozařování za pomoci zdroje záření v určité vzdálenosti, mimo tělo pacienta. Dříve se využívalo radionuklidových zdrojů záření, zde spadá především pronikavé záření gama. V současnosti jsou nahrazovány lineárními urychlovači, kde vzniká jako brzdné záření. <sup>[12]</sup> Teleterapie lze využít na ložiska, která jsou na povrchu těla, nebo těsně pod povrchem. V hloubce je můžeme použít, ale musí být možné jej využít bez zásahu důležitých orgánů, které by mohly vysoké dávky záření poškodit, jako například mícha. V lineárních urychlovačích se generuje záření v jednotkách až desítkách MeV. <sup>[4]</sup>

## **2.4 Ozáření tkáně a jeho důsledky**

Následek ozáření na živé buňky způsobí excitaci a ionizaci. V další chemické fázi můžou vyvolat tvorbu volných radikálů, poškození chemických vazeb a tvorbu nových a zničení molekul, které jsou obsaženy v DNA a RNA. Účinky mohou být buď nepřímé, kde poškození způsobuje meziprodukt radiolýzy vody (rozrušení molekul vody), nebo přímé, kde dochází k účinku přímo v molekule. U přímých účinků dojde k zasažení a zničení buněk, kterým už nezbyl čas na jejich regeneraci. V průběhu nepřímých účinků se tvoří radikály, jsou vysoce reaktivní s nepárovým elektronem ve valenční vrstvě. Ty mohou být důvodem ke kaskádové reakci v makromolekulách buněk. V ozáření tkáni je nadbytek těchto radikálů a ty způsobují změny významných molekul. Je poškozena báze, cukry a samotné vazby v DNA, kde dochází k zlomům a oddělení párů. K nepřímým účinkům dochází mnohem častěji.

Citlivost buňky je dána podle terče DNA, je-li tento terč zasažen, dojde k inaktivaci buňky a ta po určité době odumírá. Je spuštěna tzv. buněčná smrt. Zasažení buňky je kratší než  $10^{-3}$  sekundy. Předpokladem je velká dávka a velký cíl, což je buňka s velkým nebo zdvojeným jádrem. Radiosenzitivitu vykazují i kmenové buňky a rychle rostoucí buňky, kvůli intenzivnímu metabolismu, který v nich probíhá. Mají tedy mnoho kritických cílů, které lze zasáhnout zářením. <sup>[11]</sup>

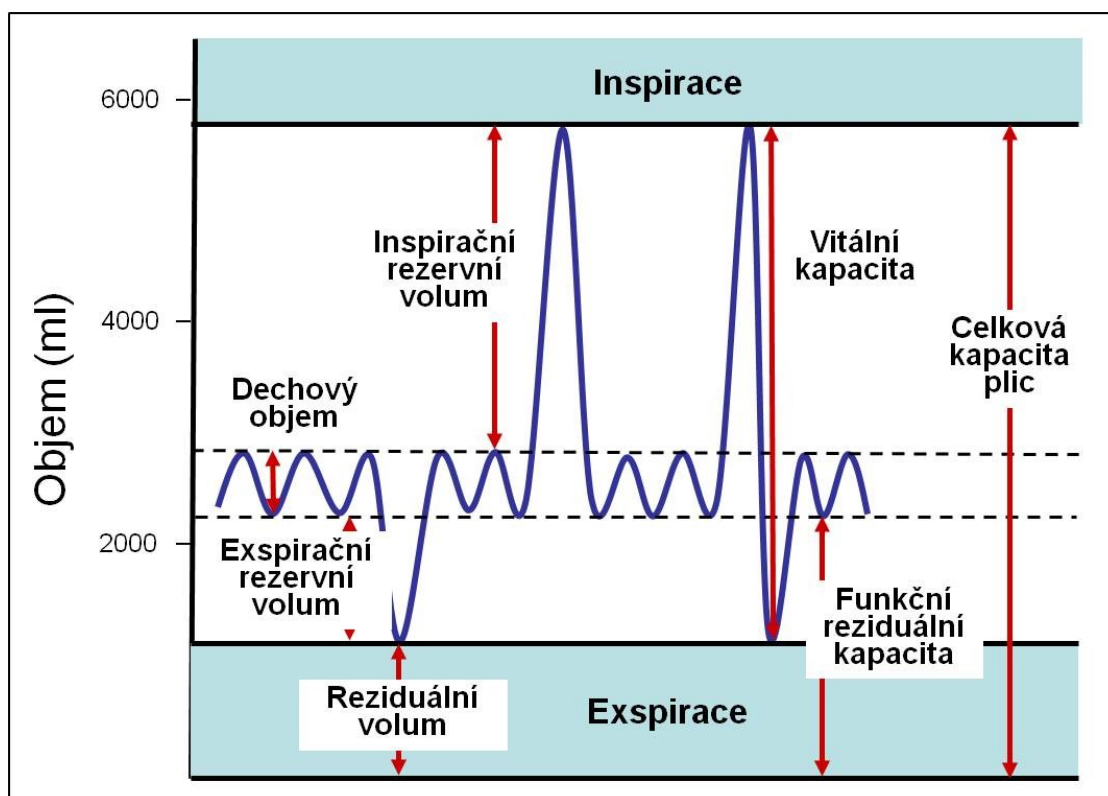


### 3 Fyziologie dýchání

Dýchání je proces, který zprostředkovává výměnu plynů mezi atmosférou a plicními alveoly, které zabezpečují tuto výměnu s krví. Mají za úkol zajistit rovnováhu mezi ventilací plic a metabolickou potřebou celého těla. V průběhu klidové ventilace je za minutu přečerpáno asi 8 litrů vzduchu. V intrapleurální štěrbině je podtlak, který se při nádechu stává více negativním, než je atmosférický tlak a tak dochází k nasátí vzduchu. V průběhu dochází k rozpínání hrudního koše, což způsobuje roztažení plic. Nádech je aktivní proces, kterému napomáhají mezižeberní svaly a bránice, která při nádechu klesá. K výdechu dochází po ochabnutí dýchacích svalů, kde se uplatňuje retrakční síla plic. Výdech je pasivní děj. Impulzy pro dýchací svaly vychází z CNS a jsou dány potřebou organismu. <sup>[13]</sup>

Dýchání zprostředkovávají mezižeberní svaly a bránice. Bránice má na dýchání největší podíl. Bránice má v klidové poloze vyklenutí směrem do dutiny hrudní. Při nádechu je oploštěná a je tak zvětšený objem dutiny hrudní. V průběhu nádechu jsou stlačeny břišní útroby a břišní stěna. Proto je inspisium dějem aktivním. Při výdechu bránice začne relaxovat a plíce spolu s hrudníkem jsou díky hmotnosti a jejich flexibilitě v klidové poloze zatlačeny dolů a napomohou tak bránici navrátit se do původního vyklenutého tvaru. <sup>[14]</sup>

V průběhu klidného dýchání je pokles bránice přibližně v průměru o 1,2 - 1,5 cm a hlubokém nádechu o 3-7cm u dospělé osoby. Tyto hodnoty se liší s každým nádechem i výdechem a jsou dány fyziologií dýchání a fyziologickými potřebami organismu. <sup>[15]</sup>



Obrázek 1: Plicní objemy

Zdroj: <http://pfyziollfup.upol.cz/castwiki2/?p=1199>

### 3.1 Nežádoucí vlivy dýchání a jejich eliminace pro radioterapii

V průběhu radioterapie je zapotřebí eliminovat veškeré pohyby pacienta na minimum, aby nedocházelo k nežádoucím přesahům ozařené tkáně. Jedním z největších problémů současné radioterapie je dýchání. Kdybychom sledovali dechovou křivku, byla by nestabilní i při naprostém klidu. Na začátku snímání je frekvence dýchání vyšší. Ve chvíli, kdy dojde k úplnému uvolnění pacienta, dojde k poklesu frekvence, ale stále není v průběhu dýchání harmonicky shodný. Z důvodu řízení dýchání fyziologickými potřebami organismu může docházet k jednorázovým hlubokým nádechům či výdechům, které se nedají předvídat. U konvenční radioterapie se proto často používá zadržování dechu, nebo je ložisko ozařováno v nejstabilnější fázi dechového cyklu pacienta a to ve výdechu. Ale ani tyto varianty nejsou nejjistější, jelikož u zadržení dechu není jisté, že jej pacient zadrží ve stejné fázi, jako při měřeních, když vznikl ozařovací plán a proto se přidávají i centimetrové lemy, které by dokázaly pokrýt plochu nádorového ložiska i při neshodách v dechovém cyklu.<sup>[9]</sup>

Radioterapie řízená respirací nabízí významný potenciál pro zlepšení ozařovaných lokalit, které pohyby dýchacích cest ovlivňují, jako jsou nádory plic, prsu, jater a dalších.

V léčbě se využívá několik způsobů, jak problémy vzniklé s dýcháním eliminovat. Mezi tyto techniky patří nucené mělké dýchání s abdominální kompresí, techniky zadržování dechu, začlenění dýchání do plánované léčby, respirační vtokové techniky a sledovací techniky dechu. Měření dýchacích pohybů mohou být prováděna buď v reprezentativním vzorku populace, nebo přímo na pacientovi před ozářením. Redukce dýchacího pohybu lze dosáhnout buď břišní kompresí, nebo technikou dýchání. Po nádechu může být dech zadržen buď aktivními technikami, ve kterých je proud vzduchu do pacienta dočasně blokován ventilem, nebo pasivními technikami, ve kterých pacient dobrovolně zadrží dech. V konvenční radioterapii se dodnes zabýval výzkum spíše tím, jak utlumit pohyby orgánů v průběhu dýchání. Břišní komprese využívá desku, která tlačí na hrudník pacienta a dokáže utlumit pohyb orgánů, který přesahuje 5mm. <sup>[16]</sup>

Integraci dýchacích pohybů v plánování radioterapie zlepšuje současná počítačová tomografie a rozvoj 3D zobrazení. S možností zobrazení kontur orgánů v nádechu a výdechu dává možnost změření rozptylu pohybu. To umožnilo zavedení real-time monitorování pacienta v průběhu ozařování. Dva externí reflektory sledují značky na pacientovi, ty odráží pohyb a poté je zpracovává software. Data zpracovaná softwarem jsou poté přenášena k ozařovací hlavici v průběhu léčby a hlídají se kritéria stanovená v průběhu plánování léčby. Jiné metody sledování pacienta v reálném čase používají různé zařízení na bázi mechanických, světelných, nebo spirometrických čidel. <sup>[16]</sup>

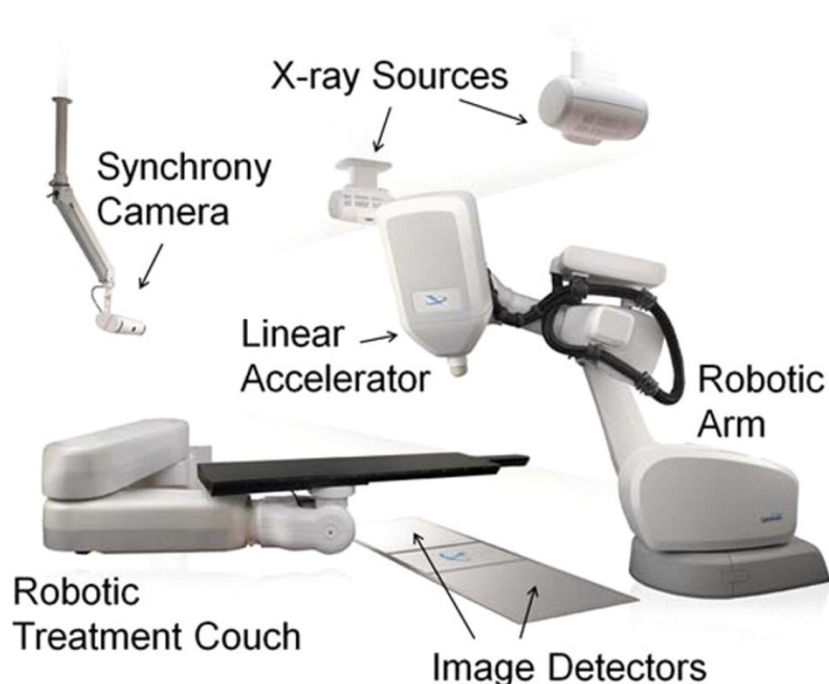
Další kategorií je sledování tumoru, který se skládá ze dvou hlavních aspektů: real-time lokalizace, a real-time přizpůsobení paprsku neustále se pohybujícího nádoru. Ze získaných dat před zahájením léčby je statisticky vyhodnocený pohyb ložiska. Amplitudy pohybu se mohou lišit z různých důvodů, například upevněním na hrudní stěnu, či respirační kapacitou pacienta, okysličením, úzkostí, osobní imobilizací nebo pozicí paže. <sup>[16]</sup>

**Tabulka 1: Hlavní studie hodnotící pohyby orgánů při volném dýchání  
v literatuře(HindawiPublishingCorporation, ISRN Pulmonology, Volume 2013, Article  
ID 519602, 13 pages, <http://dx.doi.org/10.1155/2013/519602>)**

Lokalizace	Volné dýchání	Hluboký nádech
Játra	16.7mm (5–40)	35.0mm (CC) (12–80)
Bránice	11.9mm(7–38.2)	39mm(CC) (3.1–95)
Plíce (dolní lalok)	≈10mm (0–34)	15mm (CC) (5–22)
Prostata	3.3mm (1–11)	-
Slinivka břišní	18mm (0–35)	15mm (CC) (5–22)
Ledvina	3mm (2–40)	≈20mm (4–86)
(CC): kraniokaudální směr, (x-y): min-max amplituda.		

## 4 Radiologický přístroj CyberKnife

Jedná se o přístroj k ozáření nádorových ložisek. Součástí tohoto přístroje je robotické rameno, které se dokáže pohybovat okolo stolu, na kterém leží pacient, v takovém rozsahu, že dokáže ozářit ložiska nádorů s vysokou přesností. Toto rameno dokáže kompenzovat pohyby ložiska i pacienta. Pohybu ložiska se přizpůsobuje pomocí sledování bodů ložiska a porovnává s korelačním modelem.



Obrázek 2: CyberKnife

Zdroj: <http://is-med.org/news/cyberknife/2011-05-28-88>

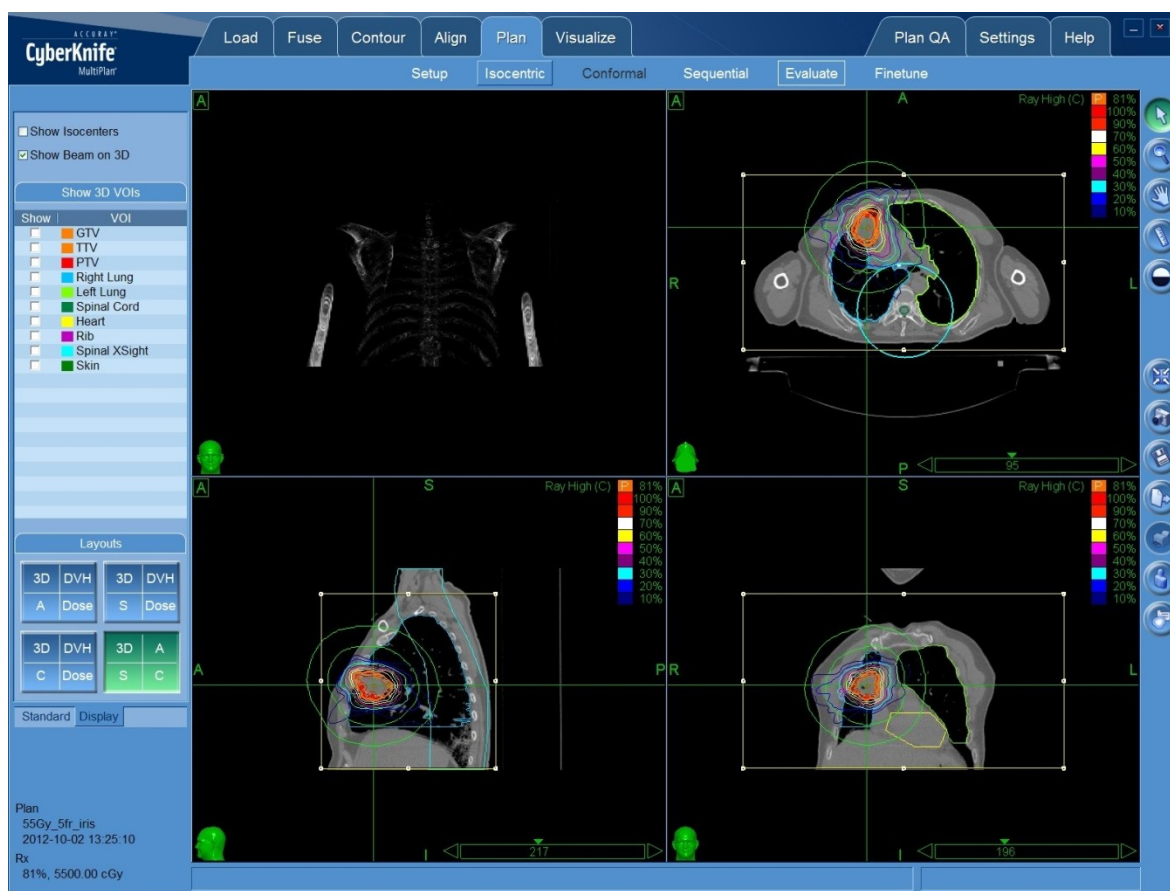
### 4.1 Získávání dechové křivky

Před zahájením léčby je zapotřebí získat referenční data. Před samotným snímáním těchto dat je pořízeno několik RTG snímků. Tyto snímky jsou snímány vždy pod úhlem 45° zleva i zprava pacienta rentgenkami, které jsou součástí přístroje CyberKnife a nacházejí se u stropu ozařovny. Je zapotřebí získat korelační model, který je dán RTG snímkem v různých fázích dýchacích pohybů, systém má poté k dispozici informaci, kde se ložisko nachází v které fázi dýchání. Díky tomuto korelačnímu modelu je možné pacienta v průběhu léčby sledovat díky vestě, která má na sobě 3 led diody. Ty jsou snímány speciální kamerou.

Poté jsou pořizovány RTG snímky v určitých intervalech, které kontrolují, zda poloha ložiska odpovídá původnímu modelu. Takto je získávána dechová křivka pacienta. [9]

## 4.2 Léčba na přístroji CyberKnife

Před zahájením léčby musí pacient podstoupit plánovací CT vyšetření a dle nutnosti i MR vyšetření s použitím kontrastních látek. V pořízeném CT vyznačí lékař kritickou oblast a ložisko, které se má ozářit. Lékař také stanovuje dávku a počet frakcí, tedy počet ozáření. Fyzik se snaží splnit požadavky a nepřekročit limitní dávky na kritické orgány. Veškeré výpočty jsou poté prováděny pomocí speciálního softwaru na výkonném počítači. Software splní zadané požadavky a nabídne co nejvhodnější variantu podle daných podmínek. [9]



**Obrázek 3: Plánovací systém přístroje CyberKnife u plicního případu**  
**Zdroj: Oddělení radiologického přístroje CyberKnife FNO Ostrava**

Přístroj CyberKnife ozařuje ložisko neisocentricky, což znamená, že každý daný svazek záření může směřovat do jiného místa. Takže každý svazek může mít své vlastní isocentrum, což u klasické radioterapie nelze. V klasické radioterapii se využívá isocentrická metoda, popřípadě plány s více isocentry, ale vždy všechny nebo několik svazků míří do stejného místa v tumoru. Tato skutečnost u přístroje CyberKnife neplatí a proto s ním lze ozařovat

s velkou přesností a s minimálním postihem zdravé tkáně. Tato podmínka je nesmírně důležitá při nádorech v oblasti páteře. Velikost lemu je poté přidávána podle podmínek a druhu nádoru.<sup>[9]</sup>

Přístroj CyberKnife je možné využít pro všechny typy nádorů, které vyžadují ozáření. Je ale zpravidla využíván tam, kde má konvenční radioterapie své limity. Nejčastěji jsou pro tuto léčbu indikovány intrakraniální ložiska, pohyblivá ložiska (pulmo, pankreas, hepar), ložiska nádorů prostaty, ložiska kritických oblastí jako je okolí páteře atd. Cílem tohoto přístroje je omezit počet frakcí na minimum, jako je tomu u karcinomu prostaty. U tohoto nádoru je stejná diagnóza jako u konvenční radioterapie, avšak je zde snížen počet frakcí až na hodnotu 5, což umožní významné zkrácení doby léčení pacienta. 1 frakce= 1 radiochirurgie, tedy nahrazení chirurgického výkonu dodáním jednorázové vysoké dávky.<sup>[9]</sup>

V průběhu celé léčby je z pacienta snímána předpokládaná a aktuální poloha ložiska a na počítači je poté viditelná přesnost celého procesu. Na 2D modelu se zobrazuje nepřesnost v milimetrech. Rozsah pro zobrazení chyby se pohybuje do pěti milimetrů. Když dojde k posunutí pacienta, rameno ukončí proces a vyšle signál obsluze, že je zapotřebí pacienta uložit jinak, toto uložení se zakládá na strukturách páteře. Špatné uložení pacienta se zjistí tak, že neshoda na předpokládaném místě a na místě aktuálním přetrvává delší dobu. Tato neshoda bývá totiž často způsobená rotací pacienta v průběhu léčby na stole a většinu problémů s korelací lze vyřešit úpravou polohy pacienta.<sup>[9]</sup>

Díky této technologii není zapotřebí přidávat rozsah, jako tomu je u konvenční radioterapie, kde není možnost sledovat pohyb pacienta a proto je zde zapotřebí rozsah rozšířit v řádech centimetrů, čímž vzniká větší poškození zdravé tkáně v důsledku ozáření. U přístroje CyberKnife se přidává lem pouhé tři milimetry, k pokrytí nepřesností a k pokrytí mikroskopického šíření nemoci, která nemusí být vidět při zakreslování.<sup>[9]</sup>

## 5 Zpracování dechové křivky

Pohyby pacienta v průběhu ozařování se ukládají do tzv. logu. Tento soubor je připravován v průběhu léčby, kdy má pacient na sobě speciální vestu s led diodami. Pohyby těchto diod jsou snímány speciální kamerou a informace o pohybu jsou zapisovány jako údaj o pohybu pacienta pro všechny 3 směry pohybu v čase.

Log začíná vždy prázdným řádkem, dále následuje informace o čase, kdy byla léčba zahájena. Tento čas je uveden ve vteřinách od půlnoci. Na třetím řádku se nachází informace o samotném pacientovi. Data jsou zde ve čtyřech sloupcích. První sloupec je informace o čase, tedy délce ozařování. Zbylé tři sloupce jsou informace o pohybu ložiska v průběhu léčby. Je ve 3D systému, jedná se tedy souřadnice X,Y,Z, odpovídající pohybům ve směru: X= Superior-inferior (od hlavy k nohám), Y=lefttright neboli laterolateral (doleva a doprava) a Z=anteriorposterir (od zad k břichu). Souřadnice se v jednom logu mohou skokově lišit, z důvodu přerušení léčby a přepolohování pacienta. Frekvence získaného záznamu je 25Hz, tedy 25 snímků za vteřinu. Čas záznamu je dán délkou samotné léčby a počet záznamů pro každého pacienta je dán počtem frakcí.<sup>[9]</sup>

Je nutné zachovat informace o pacientovi. V logu jde o rodné číslo a příjmení. Hlavním výsledkem mají být hodnoty maximálních a minimálních výchylek pohybu ozařovaného či sledovaného ložiska. Je zapotřebí odfiltrovat překmity a křivku upravit pro požadované výstupy. U těchto údajů statisticky vyjádřit jejich četnost a výskyt. Zjistit, kdy byla léčba přerušena. To lze zjistit díky delším časovým prodlevám v prvním sloupci logu. Je nutné zachovat informaci o počtu přerušení v průběhu léčby a zjistit celkový čas léčby, včetně prodlevy při přerušení i bez ní. Vizualně je pak zobrazen pohybu ložiska v prostoru.

Při použití konvenční metody je bezpečnostní lem nastaven v řádech několika centimetrů, oproti tomu přístroj CyberKnife tento lem dokáže minimalizovat na několik milimetrů. Tento lem je důležitý, aby bylo pokryto okolí nádoru, které může být posunuto pohybem pacienta. U konvenční radioterapie je tento lem větší, protože není možné sledovat a korigovat pohyby pacienta a musí být tedy pokryta celá velikost ozařovaného nádoru ve všech stádiích dýchacího cyklu. Využití navrhovaného programu pomůže zpracovávat data z dechové křivky, která jsou získávána v průběhu léčby. Měl by napomoci k zlepšení současné léčby ozařovaných nádorů.

Program navrhovaný pro zpracování dechové křivky bude sloužit pro kontrolu. Spojením sledování dechové křivky u pacientů léčených na přístroji CyberKnife



a konvenční radioterapie, může být dokázáno, jestli je bezpečnostní lem u konvenční léčby dostatečný.

Data zpracována a vizualizovaná v programu mohou a budou použita pro následnou postanalýzu kvality ozařování a návrhu optimálních metod ozařování a přístupů radiologické léčby.

## 5.1 Instalace knihoven programu Matlab

Před zahájením práce s programem, je zapotřebí nainstalovat knihovny programu Matlab, ve kterých je program vytvářen. Jsou volně stáhnutelné ze stránek mathworks: <http://www.mathworks.com/products/compiler/mcr/>.

## 5.2 Získání dat z datového souboru typu log z přístroje CyberKnif

Soubor typu log je produkován přístrojem CyberKnife. Obsahuje informace o pacientovi a průběhu jeho léčby. Data o pohybu pacienta jsou zapisována v trojrozměrném systému po celou dobu ozařování v reálném čase pod hlavičkou v souboru. V případě souboru typu log je nutné provést úpravy, aby bylo možno data z něj správně zpracovat a použít. Soubor typu log je možno vidět na následujícím obrázku:

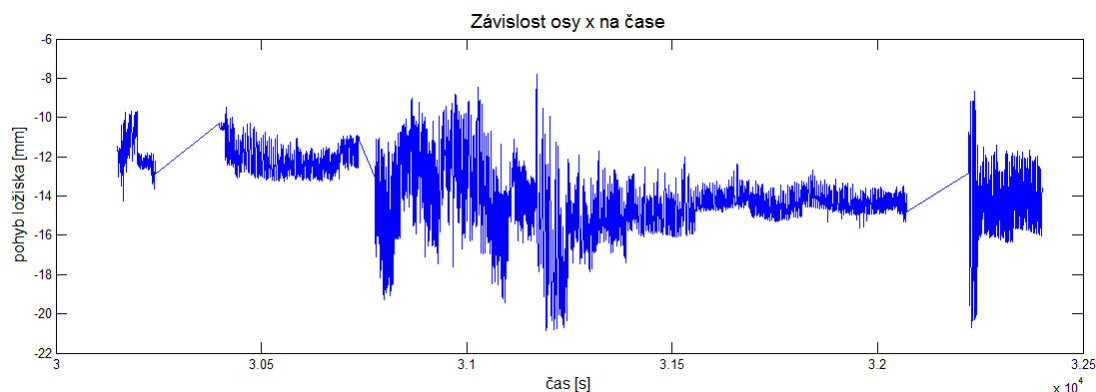
```
48708.659 DATE_ANCHOR Wed 03 Oct 2012 13:31:48
48708.659 INFO --> LoggerHandler::init: initialized id 6 with file [redacted]_03Oct12_133148/Modeler.
48708.659 INFO --> Modeler log started Wed Oct 03 13:31:48 Central Europe Daylight Time 2012
30149.248:-4.227 12.332 0.958
30149.287:-4.213 12.339 0.925
30149.325:-4.140 12.378 0.747
30149.363:-4.057 12.422 0.547
30149.401:-4.029 12.437 0.480
30149.440:-4.043 12.430 0.514
30149.478:-4.043 12.430 0.514
30149.517:-3.997 12.454 0.403
30149.556:-3.946 12.481 0.280
30149.594:-3.923 12.494 0.225
```

**Obrázek 4:Původní soubor typu log**

První řádek v souboru je prázdný. Na druhém řádku je údaj o datu a času léčby pacienta. Třetí řádek patří údajům o pacientovi. Příjmení pacienta a rodné číslo je skryto úmyslně z důvodu zachování anonymity pacientových osobních dat. Na pátém řádku se začínají načítat data v průběhu léčby.

Při zobrazení osy x bez úprav je vidět velké množství nepřehledných dat, rovné úseky, které znamenají přerušení. Je tedy snadno okem zjištětné, kolikrát byla léčba přerušena. Tyto úseky nejsou při zobrazení samotné osy žádoucí. Křivku je zapotřebí vyhladit, ale zachovat hodnoty extrémů. Je možno si povšimnout úseků, které odpovídají

klidovým stádiím léčby, kdy byl pacient téměř v klidu a jeho dechové amplitudy jsou téměř totožné, tyto poznatky lze vidět hlavně v druhé polovině léčby.



**Obrázek 5: Osa x v závislosti na čase**

### 5.3 Zjištění informací z hlavičky logu

Při stisknutí tlačítka načíst soubor a vybrání správného souboru typu log, se spustí funkce s názvem „hlavicka“, která načte soubor a začne jej prohledávat, aby odstranila veškeré znaky, které by v průběhu dalších procesů Matlabu způsobovaly potíže. Jsou odstraněna zejména interpunkční znaménka (., \_ , /) a upravený soubor je uložený jako textový soubor.

```

data_upravene – Poznámkový blok
Soubor  Úpravy  Formát  Zobrazení  Nápověda

48708.659 DATE ANCHOR Wed 03 Oct 2012 13 31 48
48708.659 INFO --> LoggerHandler  init  initialized id 6 with file  03Oct12 133148 Modeler.
48708.659 INFO --> Modeler log started Wed Oct 03 13 31 48 Central Europe Daylight Time 2012
30149.248 -4.227 12.332 0.958
30149.287 -4.213 12.339 0.925
30149.325 -4.140 12.378 0.747
30149.363 -4.057 12.422 0.547
30149.401 -4.029 12.437 0.480
30149.440 -4.043 12.430 0.514

```

**Obrázek 6: Upravená data bez závadných znaků.**

Takto upravený soubor podrobíme funkci se skenováním textu (textscen), která vyhledává textové znaky, rozloží je na slova a každé z nich uloží do samostatné buňky matice, ze které poté tyto znaky použijeme do struktury, podle jejich umístění v tomto souboru. Jelikož se jedná o znaky, které jsou odděleny mezerou, nezáleží na délce slova, které bude do struktury vypsáno. Proto nebude žádný problém při vypisování různých délek příjmení pacienta.

	1	2	3	4	5	6
1	48708.659					
2	DATE					
3	ANCHOR					
4	Wed					
5	03					
6	Oct					
7	2012					
8	13					
9	31					
10	48					
11	48708.659					
12	INFO					

**Obrázek 7: Tabulka s datovým souborem, na který byla použita funkce textscen**

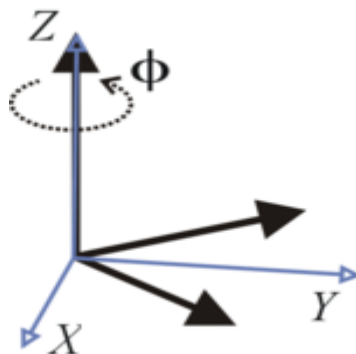
Po získání těchto důležitých údajů jsou následně z původního souboru odstraněny 4 první řádky, aby se do matice uložila pouze data dechové křivky. Tedy čas, osa x, osa y a osa z.

## 5.4 Přepočítání úhlu křivky

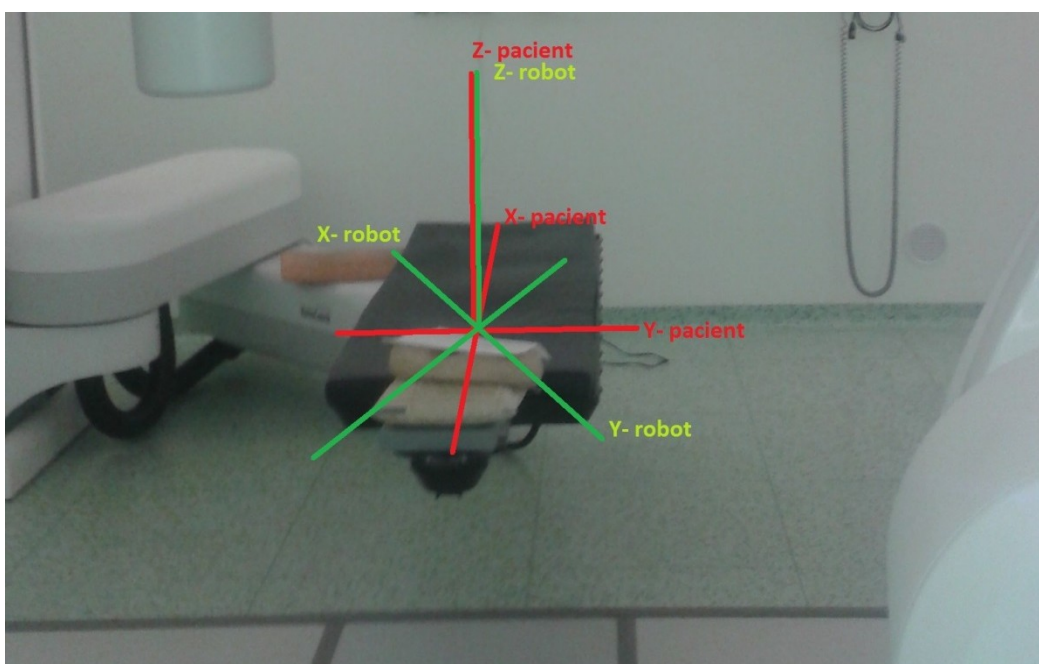
Data jsou zaznamenávána v průběhu léčby vzhledem k robotovi, který ozařování provádí. Z toho důvodu jsou data vyrotovaná o  $-45^\circ$  vůči pacientovi. Je proto potřeba před samotným vykreslením a dalším zpracování dat data přepočítat o  $+45^\circ$ .

$$x_{(45^\circ)} = (x * \cos 45^\circ) - (y * \sin 45^\circ) = (x * 0,7071) - (y * 0,7071)$$

$$y_{(45^\circ)} = (y * \cos 45^\circ) + (x * \sin 45^\circ) = (x * 0,7071) + (y * 0,7071)$$



**Obrázek 8: Rotace osy x a y o 45°**  
**Zdroj:** [http://www.wikiwand.com/en/Euler\\_angles](http://www.wikiwand.com/en/Euler_angles)



**Obrázek 9: Popis os vůči robotovi a vůči pacientovi**  
**Zdroj:** Oddělení radiologického přístroje CyberKnife FNO Ostrava

## 5.5 Čas a počet přerušení

Každý řádek logu včetně hlavičky je opatřen časovým údajem. Čas je měřený v sekundách od půlnoci, proto je zapotřebí tato nepřehledná čísla přepočítat, aby je bylo možné zobrazit jako čas. Časový údaj je potřebné zobrazit jako informační údaj v tabulce programu, kde se zobrazuje čas z hlavičky souboru, který je uložen ve struktuře dat a není nijak přepočítáván, jelikož tam je čas už přepočítán. Je pouze načtený ze struktury. Přepočítávány jsou až časy, které se nacházejí u samotných hodnot. Celý časový sloupec je podroben dělení ve funkci s názvem „delení“, které zjišťuje počet hodin a zapíše tuto

hodnotu do samostatného sloupce. Zbytek je opět vydělen, a tím je zjištěn počet minut a opět je samostatně zapsán do sloupce, stejně tak i počet sekund a milisekund.

	1	2	3	4	5	6	7	8	9
1	8	22	29	14.8800	-11.7089	5.7310	0.9580	1	
2	8	22	29	17.2200	-11.7039	5.7459	0.9250	1	
3	8	22	29	19.5000	-11.6799	5.8251	0.7470	1	
4	8	22	29	21.7800	-11.6523	5.9149	0.5470	1	
5	8	22	29	24.0600	-11.6431	5.9453	0.4800	1	
6	8	22	29	26.4000	-11.6481	5.9304	0.5140	1	
7	8	22	29	28.6800	-11.6481	5.9304	0.5140	1	
8	8	22	29	31.0200	-11.6325	5.9799	0.4030	1	
9	8	22	29	33.3600	-11.6155	6.0351	0.2800	1	
10	8	22	29	35.6400	-11.6085	6.0606	0.2250	1	
11	8	22	29	37.9800	-11.6127	6.0450	0.2580	1	
12	8	22	29	40.2600	-11.6120	6.0499	0.2470	1	

**Obrázek 10: Výstupní data po kompletní úpravě**

V rámci léčby může docházet k přerušení procesu, která jsou zaznamenány jako trhlina v časové posloupnosti. Tato přerušení bývají nejčastěji z důvodu přenastavení pacienta, u kterého došlo například pohybem k posunutí a robot potřebuje pacienta dostat opět do správné polohy, aby ložisko ozařoval správně.

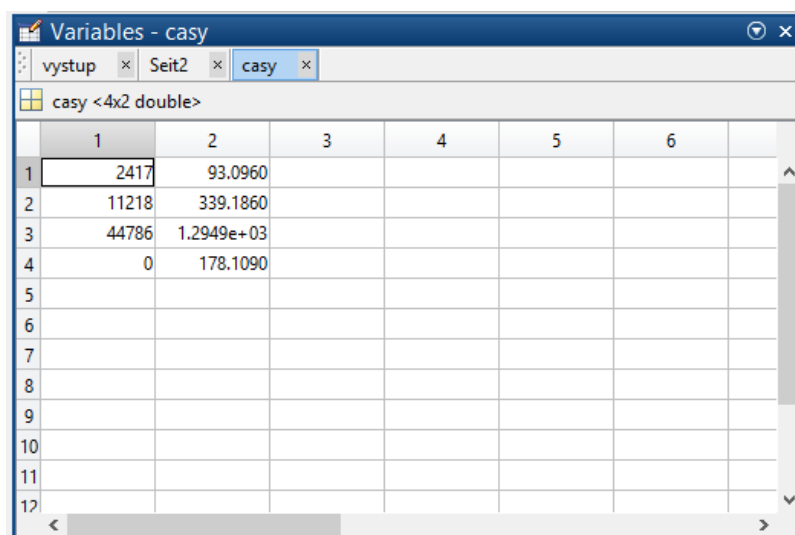
Pro zjištění počtu přerušení je spuštěn řetězec, který prozkoumává časy v prvním sloupci matice a porovnává jeden po druhém. Když zjistí rozdíl v čase větší než 0,1s dojde k zapsání informace, že nastalo přerušení. Údaj o tom, do kterého intervalu data náleží, je vypsán do prázdného sloupce za každý řádek, je tedy následně dohledatelné ve které části procesu se data nachází a taktéž i kolikrát byla léčba z jakýchkoli důvodů přerušena.

	1	2	3	4	5	6
2412	3.0242e+04	-12.7759	2.7916	4.0360	1	
2413	3.0242e+04	-12.8020	2.8037	3.9300	1	
2414	3.0242e+04	-12.8325	2.8171	3.8110	1	
2415	3.0242e+04	-12.8508	2.8256	3.7380	1	
2416	3.0242e+04	-12.8692	2.8341	3.6640	1	
2417	3.0242e+04	-12.8918	2.8440	3.5710	1	
2418	3.0398e+04	-10.2749	3.5433	3.9300	2	
2419	3.0398e+04	-10.2862	3.5843	3.8630	2	
2420	3.0398e+04	-10.2961	3.6196	3.8080	2	
2421	3.0398e+04	-10.3060	3.6550	3.7520	2	
2422	3.0398e+04	-10.3187	3.6974	3.6830	2	
2423	3.0398e+04	-10.3307	3.7420	3.6120	2	

**Obrázek 11: Uložená data v pořadí čas, osa x, osa y, osa z a interval**

Ukládá se číslo řádku, na kterém k přerušení došlo a současně je zapsán čas úseku, tedy délka časového intervalu, než došlo k přerušení.

Tato data jsou následně zobrazována v tabulce, která je v programu umístěna, aby byla tato data přehledná a dále zpracovatelná. Kupříkladu jsou sečteny a pomocí funkce jsou převedeny na skutečný tvar hodin a v programu zobrazeny jako údaj o délce léčebného cyklu bez mezer.



	1	2	3	4	5	6
1	2417	93.0960				
2	11218	339.1860				
3	44786	1.2949e+03				
4	0	178.1090				
5						
6						
7						
8						
9						
10						
11						
12						

**Obrázek 12:Řádek a hodnota přerušení léčby jak ji uloží program**

## 5.6 Struktura

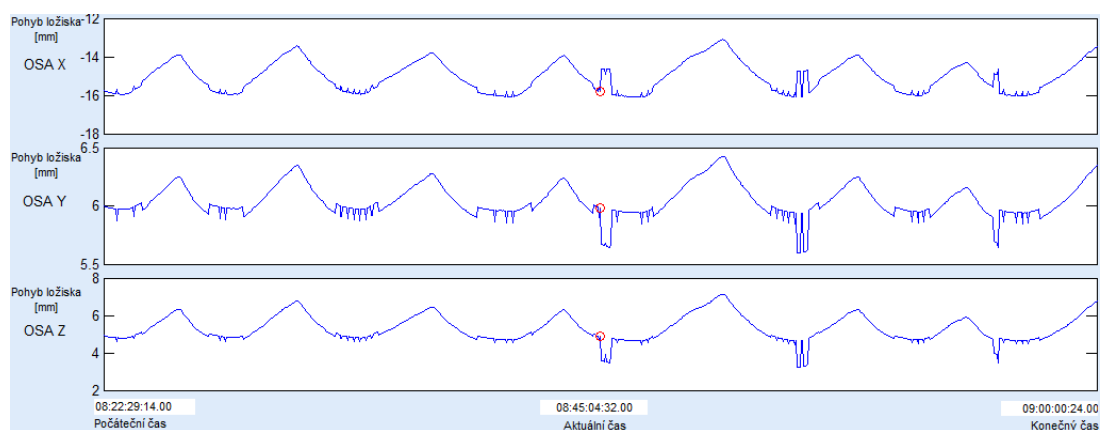
Data vyčtená ze souboru typu log jsou po veškerých úpravách uložena do přehledné struktury, ve které jsou připravena pro další zpracování, využití a zobrazení. Zvlášť je uloženo příjmení pacienta, jeho rodné číslo, datum léčby, čas zahájení léčby, matice s daty os x, y, z, počet přerušení a jejich časy.

Field	Value	Min	Max
Prijemni	[redacted]		
Rodne_Cislo	[redacted]		
Datum_lecby	'Wed 03 Oct 2012'		
Cas_zahajeni_lecby	'13:31:48'		
DataXYZ	<49401x8 double>	-20.84...	59.9400
Pocet_preruseni	3	3	3
Casy	[2417 93.0960;11218 3...]	0	44786

**Obrázek 13:**Data vyčtená ze souboru stručně uložena do struktury

## 5.6 Filtrování dat

Data zapisována do logu jsou o frekvenci 25Hz. Zápis těchto dat v tak malé časové odchylce způsobí značný šum. Pro lepší zobrazení křivek je důležité vyfiltrovat data od všeho, co není vyžadováno. Je ovšem zapotřebí ponechat veškerým maximům a minimům jejich hodnotu. Na následujícím obrázku je vidět načtená dechová křivka z původních dat před filtrováním. Jsou jasné přesahy a šum, které by pro následující funkce pro hledání extrémů činily značné potíže a proto je nutné tyto nedostatky odstranit.



**Obrázek 14:** Načtená přibližná křivka z původních dat

Pro úpravu těchto signálů byl zvolen FIR filtr, který je v Matlabu definován jako filtr FIR2. Spolu s IIR filtry jsou FIR filtry základní dva typy digitálních, nejpoužívanějších filtrů pro zpracování signálu aplikací. FIR je zkratka pro „Finite Impulse Response“, v překladu je to filtr s konečnou impulzní odezvou. Obvykle je impulsní odezva konečná, protože u FIR filtrů neexistuje žádná zpětná vazba. Nedostatek zpětné vazby zaručuje, že impulzní odezva bude konečná. Z tohoto důvodu je termín konečná impulzní odezva synonymem pro „téměř bez zpětné vazby“. Lineární fáze filtru je zpožděna od vstupního signálu, ale nenarušují jeho fázi. Jsou vhodné pro použití snížení nebo zvýšení vzorkovací frekvence. Koeficienty jsou menší nebo rovny jedné. Jsou snadno proveditelné, ale náročné na výpočty, a tedy na paměť počítače. <sup>[16]</sup>

Fir2 navrhuje vzorkovací frekvence na bázi digitálních FIR filtrů s frekvenční odezvou libovolného tvaru. Vrací řádkový vektor, který obsahuje  $n + 1$  koeficientů příkazu  $n$  z FIR filtru. Frekvenční charakteristiky a velikosti těchto filtrů se shodují a jsou dány vektory  $f$  a  $m$ , ty musí být stejně dlouhé. <sup>[17]</sup>

- $f$  je vektor frekvenčních bodů, určuje tedy, na jakých frekvencích bude docházet ke změně přenosu. Je dána v rozmezí od 0 do 1, kde 1 odpovídá polovině vzorkovací frekvence (Nyquistova frekvence). První bod  $f$  musí být 0 a poslední bod 1. Frekvence body musí být ve vzestupném pořadí.
- Vektor  $f$  použitý v programu:

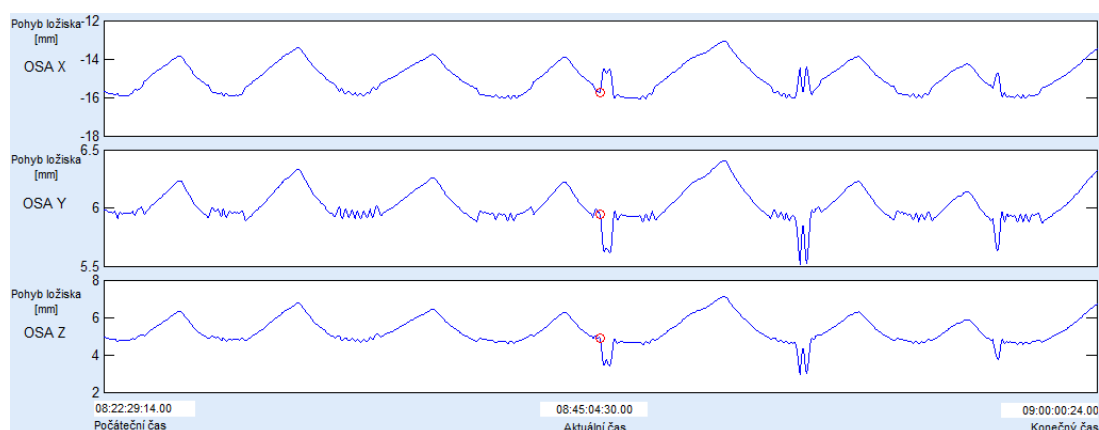
$$f = [0 \ 0.15 \ 0.15 \ 1]$$

- $m$  je vektor obsahující požadované velikostní odpovědi na místech uvedených v  $f$ . Je to přenos filtru na frekvencích  $f$ . Maximální přenos je v hodnotě 1, filtr tak neovlivní průchozí signál. Minimum je v hodnotě 0, v této hodnotě je signál maximálně potlačován. <sup>[18]</sup>
- Vektor  $m$  použitý v programu:

$$m = [1 \ 1 \ 0 \ 0]$$

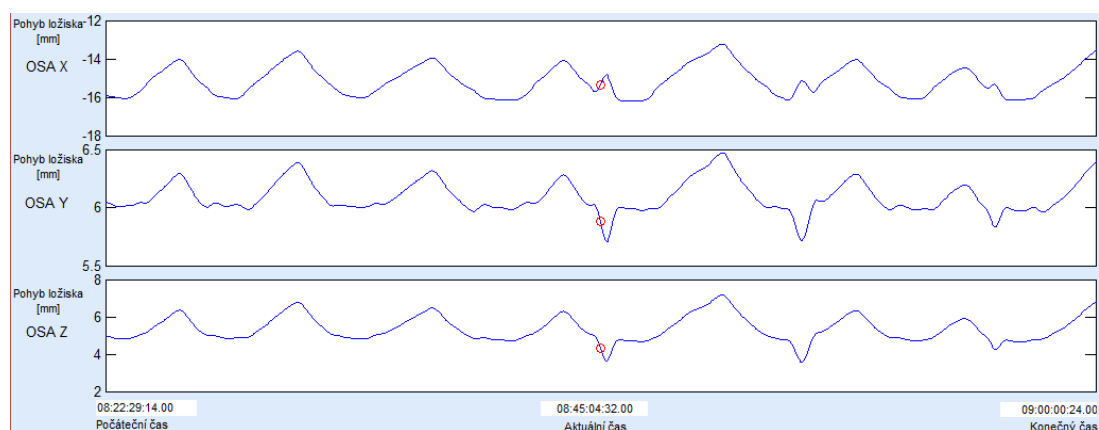
Tento filtr byl zvolen pro jeho vlastnosti a vizualizaci. Zadanými hodnotami pro hodnotu  $m$  je vytvořena ideální dolní propust, která v dolních frekvencích signál propustí a na horních frekvencích signál potlačuje. Návrh filtru (`fir2(30,f,m)`) s délkou impulzní charakteristiky 30, byl zvolený parametr jako kompromis. Čím větší je tato hodnota, tím lépe filtr filtruje, ale je o to náročnější na výpočet. Filtr je nastavený podle konstant  $f$  a  $m$ . Poté je spuštěn příkaz „`filtfilt`“, tedy filtrování bez zpoždění signálu. Je tak zabráněno zbytečným výpočtům a proces se urychlí. Po spuštění filtru nám zůstaly zachovány hodnoty extrémů a časová osa. Došlo k odstranění všech nepotřebných výkyvů v signálu.





**Obrázek 15: Použití FIR2 filtru s nastavením vektoru  $f=[0 \ 0.5 \ 0.5 \ 1]$**

Porovnáním křivky na obrázku č.14, kde je původní nefiltrovaná křivka s obrázkem č.15, kde byl nastaven vektor  $f=[0 \ 0.5 \ 0.5 \ 1]$  je viditelné vyhlazení signálu. Je patrné, že vyhlazení není dostatečné a křivka je stále plná nadbytečných překmitů, které ji znehodnocují v jejím zobrazení. Na následujícím obrázku č.16 je vidět křivka s nastavením vektoru  $f=[0 \ 0.15 \ 0.15 \ 1]$ . Zde je již křivka krásně vyhlazená a je tedy patrné, že je nastavení filtru správné.



**Obrázek 16: Použití FIR2 filtru s nastavením  $f=[0 \ 0.15 \ 0.15 \ 1]$**

## 5.7 Extrémy

Extrémy v datovém souboru zastupují funkci nádechu a výdechu, tedy vrcholů křivky. Jejich správné načtení je důležité pro další funkce programu, jako počet hlubokých nádechů, počet hlubokých výdechů, úplné maximum či minimum v úseku.

Extrémy jsou hledány pomocí matematické funkce s druhou derivací. Tato funkce porovnává bod po bodu každé osy a hledá extrémy, v případě, že prohledávaná hodnota není extrém, zapíše se do řádku jako nula, v opačném případě, je-li tedy extrém nalezen, zapíše se

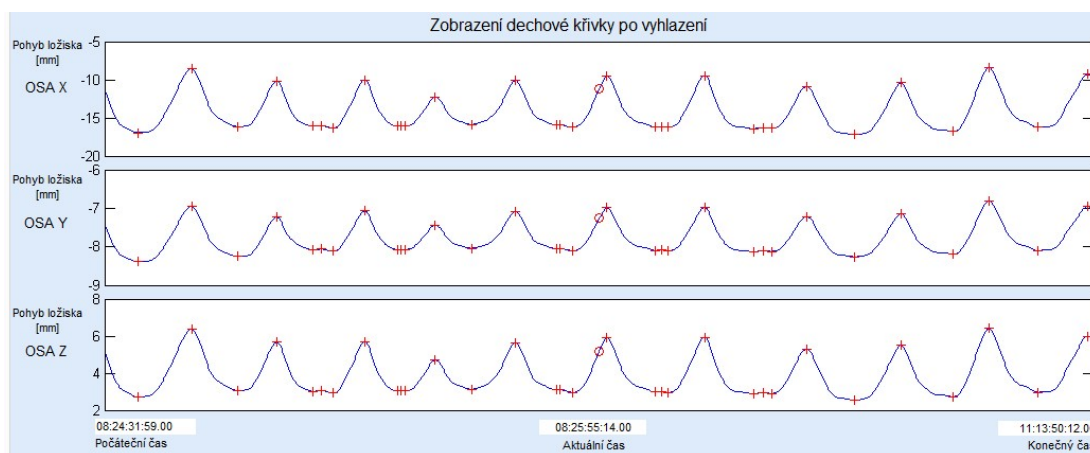
jeho hodnota. V tomto případě jsou tedy extrémy zapisovány do jednoho řádku a nelze tedy rozpoznat, jedná-li se o maximum, či minimum respektive extrémy nejsou rozděleny na maxima a minima. V programu jsou extrémy zobrazovány jako červené křížky v osách.

	4106	4107	4108	4109	4110	4111	4112	4113	
1	0	0	0	-12.7120	0	0	0	0	0
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									

**Obrázek 17: Hledání extrémů, hodnotu si zachová pouze extrém.**

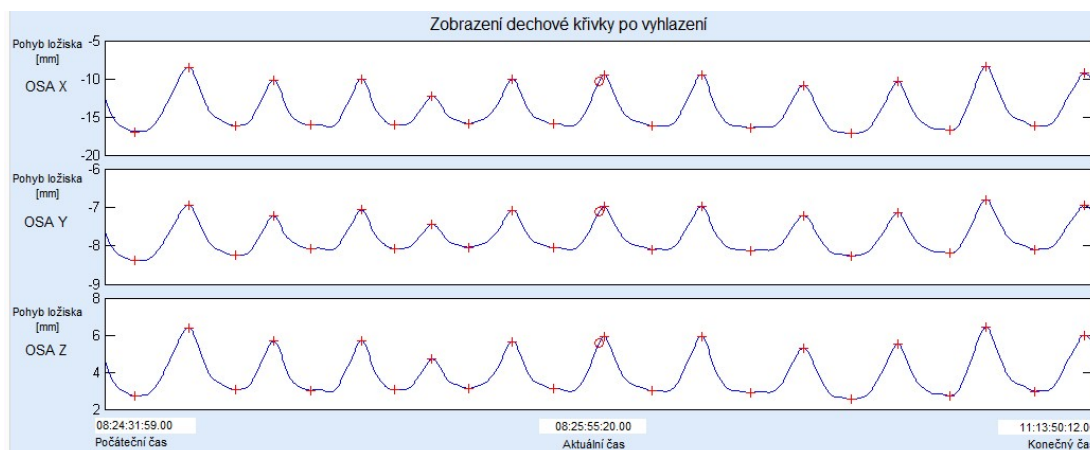
Z důvodu velkého počtu dat často nastává situace, že je jako extrém považována hodnota, která jen nepatrně postoupila, ale není vrcholem nádechu či výdechu, proto je v programu funkce (MaxMin), která porovnává po sobě jdoucí vyhledané extrémy a hledá veškeré, které se od sebe liší o hodnotu 30. Tak je zaručeno, že po sobě jdoucí extrémy budou lépe odpovídat skutečným nádechům a výdechům. Je také zvýšená pravděpodobnost, že po maximum bude následovat minimum a po něm opět maximum. Koeficient 30 je nastavován tak, aby chyba byla co nejmenší v závislosti na testovaných souborech.

Na následujícím obrázku jsou vyhledávané extrémy bez použité funkce MaxMin, tedy bez nastaveného minimálního rozdílu mezi dvěma sousedními extrémy. Je patrné, že tato funkce nefunguje správně, jelikož nám nepomůže nalézt po sobě jdoucí nádech a výdech.



**Obrázek 18: Načtená dechová křivka bez funkce MaxMin**

Na následujícím obrázku je načten stejný úsek dechové křivky, jako na obrázku předcházejícím. Nastavený koeficient 30 pro sousedící extrémy je zde dostatečný. Ve vrcholu každého nádechu a výdechu je pouze jeden křížek označující extrém. Toto nastavení je nezbytné pro budoucí zpracování dat. Hodnoty extrémů jsou totiž klíčové.



**Obrázek 19: Použití funkce MaxMin s rozdílem mezi dvěma extrémy 30**

Jelikož funkce pro extrémy byly řádkové, jsou data transponována do sloupců, aby byl zachován původní systém struktury a data se do této struktury mohla opět uložit. Po transponování dat je spuštěn jednoduchý filtr, který hledá všechna nenulová data, ale nechá je připsána jejich časovému údaji, aby se na časové ose zobrazovala správně.

Další funkcí v programu je funkce pojmenovaná `extremmax_min`. Tato funkce vyplňuje tabulku v programu o další údaje, jako je maximální nádech a maximální výdech pro každý úsek. Je to hodnota, která vykazuje naprostý extrém v daném úseku, tedy naprosté

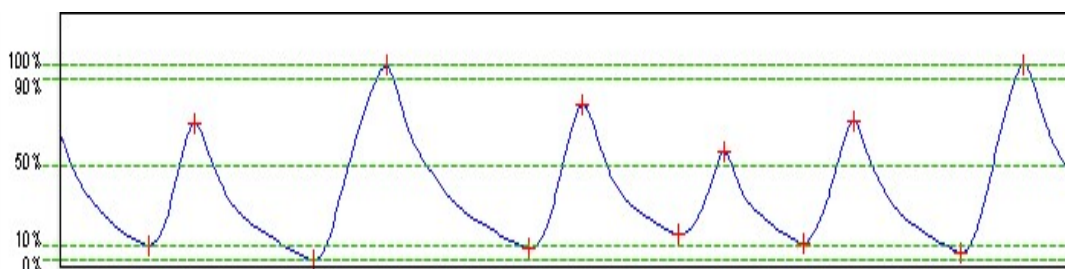
maximum a naprosté minimum. Tato funkce prohledává jednotlivé nenulové prvky matice, dokud není nalezená maximální popřípadě minimální hodnota ze všech extrémů v daném úseku.

Aby v následujících výpočtech nedělaly problémy nulové hodnoty, jsou nejprve všechny nulové hodnoty převedeny na typ NaN, které Matlab ve výpočtech přeskočí a tudíž ve výpočtech nebudou figurovat.

Po nalezení všech extrémů je použita funkce, která v hodnotách extrémů vyhledává hluboké nádechy a hluboké výdechy. Pro tento účel je zapotřebí data nejprve upravit, z toho důvodu, že funkce je nastavena na práci s procenty a data, která logu obsahuje, nabývají kladných i záporných hodnot v jednom souboru. Poté jsou data převedena do intervalu v rozmezí  $<0;1>$ , čímž nebudou dělat problémy rozdílná znaménka v souboru. Pro normalizaci dat do požadovaného intervalu je využitý vzorec:

$$x_i' = \frac{x - \min(x_i)}{\max(x_i) - \min(x_i)}$$

Normalizace zaručí správné fungování následujících funkcí. Napřed je použita funkce, která z načtených extrémů vyfiltruje maxima a minima do samostatného sloupce. Takže vzniká 6 sloupců. Zároveň je zde přidána funkce, která v matici hledá hodnoty extrémů, které jsou větší než 90% a menší než 10% křivky. Vysvětlení této funkce je vidět na obrázku.

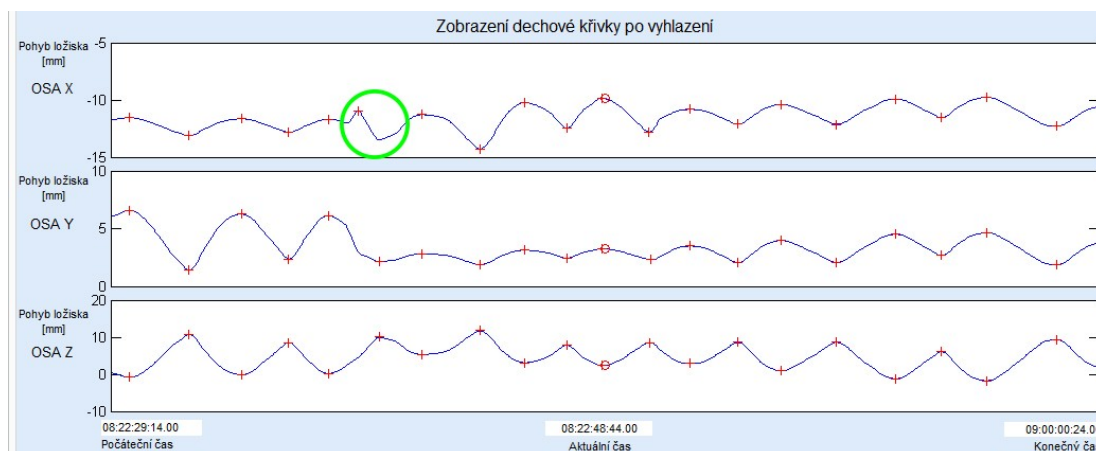


**Obrázek 20: Ilustrační graf pro vysvětlení funkce pro hledání hlubokých nádechů a výdechů**

V křivce jsou nalezeny extrémy, a následně je křivka procentuálně rozdělena. Jako hluboké nádechy jsou považovány extrémy v horních 90% a jako minima všechny extrémy, které leží v prvních 10%. Všechna data jsou poté vypsána do samostatné struktury, aby je bylo možné použít v dalších částech programu.

V datech extrémů bývají v malém procentu nalezených dat nepřesnosti, které vznikají filtrací signálu a zápisem pro hledání maxim a minim. Nastavení parametrů je takové, aby chyby byly co nejmenší. Chyby byly spočítány při závěrečném testování.

Výsledná chyba je menší než 5% a pro použití tohoto programu je velikost této chyby akceptovatelná. Na obrázku č.21 je zobrazená chyba vidět v zeleném kroužku. Po maximum nastává minimum, které ale program neakceptuje.



**Obrázek 21: Chyba v extrému**

## 5.8 Program v grafickém rozhraní GUI

Po spuštění programu se otevře grafické rozhraní vytvořené v programu Matlab v nástroji GUI (graphical user interface). V tomto rozhraní je tlačítko pro načtení souboru, kde je možno hned vedle v rámečku vidět cestu k souboru. Při spuštění načtení souboru se spustí veškerý sled funkcí popsaných doposud. Při načtení dat ze souboru log je zapotřebí stisknout šipku banneru, aby se data zobrazila správně, protože načtení křivky je závislé na funkci banneru. Ta synchronizuje zobrazení pro všechny tři osy.

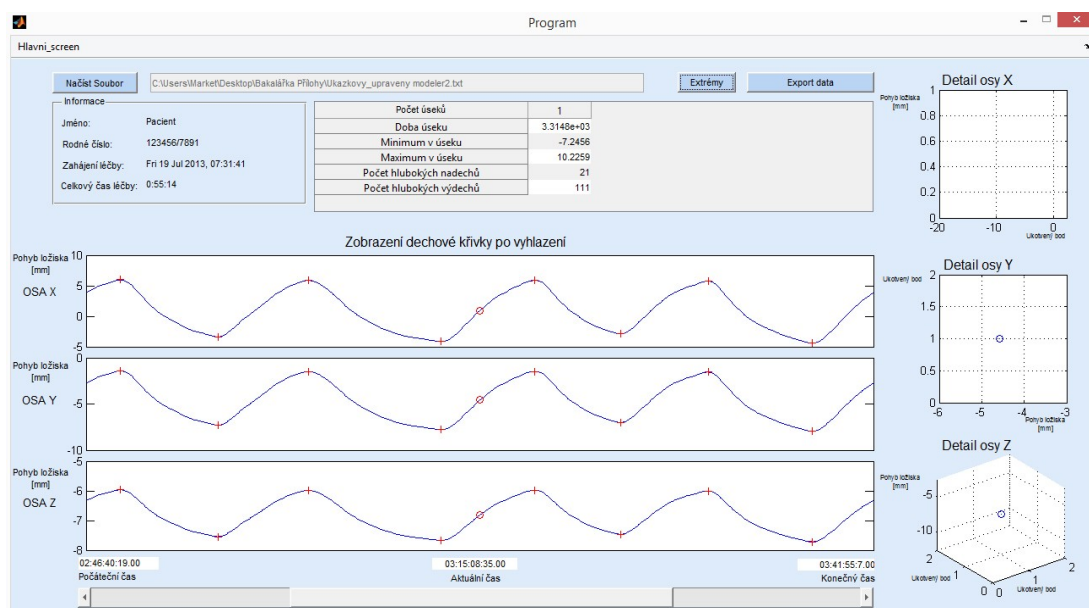
V rámečku pod tlačítkem načtení souboru je tabulka s informacemi o pacientovi a léčbě. Vyčteme zde příjmení pacienta, rodné číslo a zahájení léčby v podobě data zahájení léčby a času. Poslední údaj zobrazuje dobu léčby bez přerušení, tedy součet časů všech úseků.

V tabulce jsou poté vypsány informace vyčtené z dechové křivky. Podle počtů sloupců je zjistitelný počet úseků. V případě více úseků se objeví pod tabulkou banner pro posunutí. První údaj vypisuje dobu úseků v sekundách, tedy ve tvaru v jakém jsou načítány samotným robotem. Pod tímto údajem jsou hodnoty vyčteny pomocí funkce extrém\_max\_min. Tedy naprosté maximum a minimum v úseku. A počet hlubokých nádechů a výdechů samostatně, které jsou v programu nastaveny jako 10% hodnota od maxima či minima v daném úseku.

Pod rámečkem a tabulkou jsou zobrazovány samotné dechové křivky pro každou osu zvlášť. V x-ové ose se navíc zobrazují šedě a čerchovaně přerušení, v místě, kde došlo k zastavení léčby a přenastavení polohy pacienta. Červené body v osách jsou extrémy. Rozsahy na osách každého grafu jsou automaticky programem upravovány podle potřeby, aby byly grafy přehledné.

Posun banneru je nastaven na velmi pomalý krok, aby bylo možno se dostat ke každému potřebnému bodu grafu. Čas zobrazovaný pod postníkem je vlevo čas, který odpovídá času zahájení léčby, uprostřed je čas, který udává, kde se v grafu nachází červený kruh. Vpravo je čas ukončení léčby. Zde si lze všimnout rozdílu zahájení léčby z rámečku a z hodnot naměřených při ozařování, která jsou způsobena nesynchronními údaji počítačů, které data zapisují.

Grafy napravo se zobrazují jako bod v aktuálním čase. Jsou vhodné pro lepší odečtení hodnoty. Pohybují se podle pohybu v grafu. U osy X je pohyb nahoru a dolů, podle skutečného pohybu a u osy Y se pohybuje doleva a doprava.

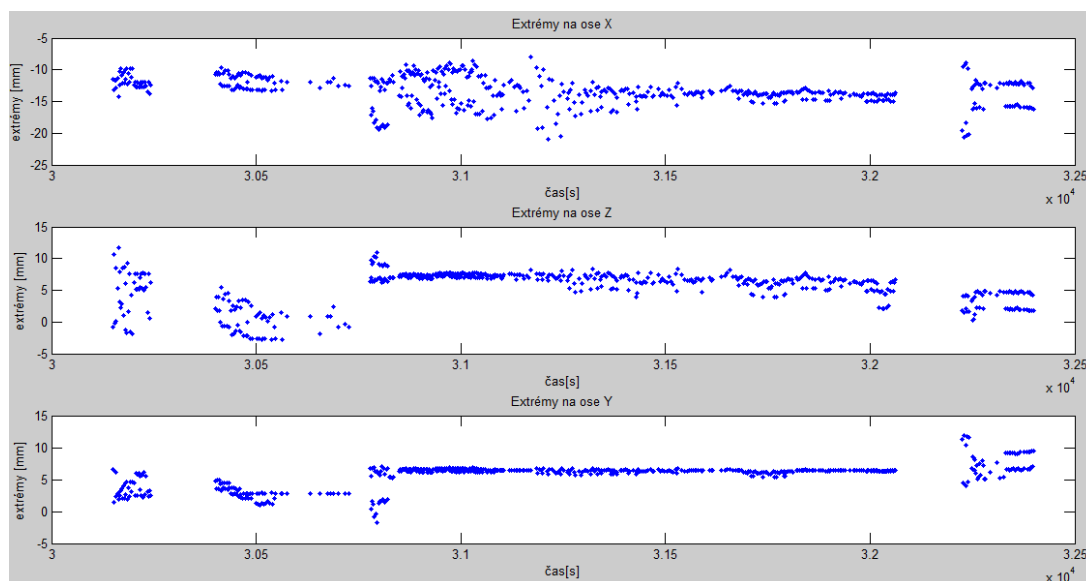


**Obrázek 22: Ukázka načteného souboru v programu**

## 5.9 Zobrazení extrémů

Vedle rámečku s cestou k souboru je tlačítko pro zobrazení maxim a minim. Toto tlačítko spustí funkci, která zobrazí veškeré extrémy, které byly vyhledány pomocí funkce extrémy po upravení dat. Tyto extrémy jsou zobrazovány ve všech třech osách i v hlavním programu pro každou křivku jako červené křížky. Při stisknutí tlačítka extrémy jsou vidět

pouze extrémy pro každou osu bez původní křivky, v bodovém grafu. Takto zobrazené extrémy jsou mnohem přehlednější, než kdyby v nich byla zobrazená původní křivka.

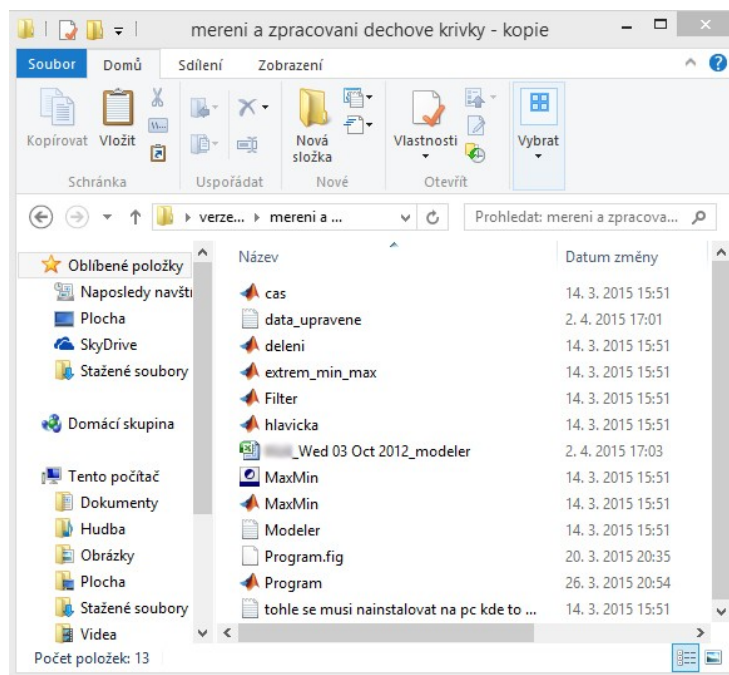


**Obrázek 23: Zobrazení extrému po stisknutí tlačítka „extrémy“**

## 5.10 Export dat

Extrémy, které jsou vyčištěny od nulových hodnot, je spolu s časem, který těmto hodnotám náleží, možné exportovat do souboru typu .xlsx, který je zpracovatelný programem Microsoft Excel. Export je prováděn pomocí tlačítka export v programu, do složky ve které se program nachází. Podoba souboru je: (příjmení pacienta\_datum zahájení léčby\_modeler.xlsx). Při exportu je vytvořena tabulka s nadpisy sloupců, obsahující veškeré vypsané extrémy, bez nulových hodnot.





**Obrázek 24: Umístění exportovaného souboru**

Při otevření vytvořeného souboru jsou data zobrazena v pořadí hodiny, minuty, sekundy, milisekundy, extrémy na ose X, extrémy na ose Y a extrémy na ose Z. Tato data jsou připraveny pro další výpočty. Rozsah souboru je dán počtem extrémů v původním souboru vytvořeném v Matlabu při zpracování.

	A	B	C	D	E	F	G	H
1	Hodiny	Minuty	Sekundy	Milisekundy	Extrémy na ose X	Extrémy na ose Y	Extrémy na ose Z	
2	8	22	30	12,6	-11,55430288	6,578768636	-0,825069148	
3	8	22	32	31,02	-13,12165946	1,413314942	10,67642192	
4	8	22	34	33,36	-11,64654296	6,275022645	-0,148311782	
5	8	22	36	24,12	-12,83456641	2,358890315	8,571250566	
6	8	22	37	56,46	-11,69210339	6,12529058	0,185095197	
7	8	22	41	35,76	-11,28538494	2,807705597	5,347205392	
8	8	22	43	49,74	-14,29741885	1,86964091	11,73062886	
9	8	22	45	36,42	-10,22549152	3,13757415	3,099630045	
10	8	22	47	13,8	-12,47525854	2,436967116	7,868273702	
11	8	22	48	37,38	-9,848706812	3,25506969	2,300523118	
12	8	22	52	3,78	-10,82646495	3,48924215	2,864153498	
13	8	22	53	55,14	-12,11898319	2,065385902	8,572517368	
14	8	22	55	37,2	-10,39713185	3,967512004	0,956199608	
15	8	22	57	47,1	-12,13041243	2,055517334	8,617002295	
16	8	23	0	6,06	-9,910180363	4,510442235	-1,208665348	
17	8	23	1	52,32	-11,54923413	2,684946731	6,073919599	
18	8	23	3	38,64	-9,783908287	4,640920285	-1,747124834	
19	8	23	6	22,62	-12,28418284	1,88531228	9,297251212	

**Obrázek 25: Exportovaná tabulka zobrazená v programu MicrosoftExcel**

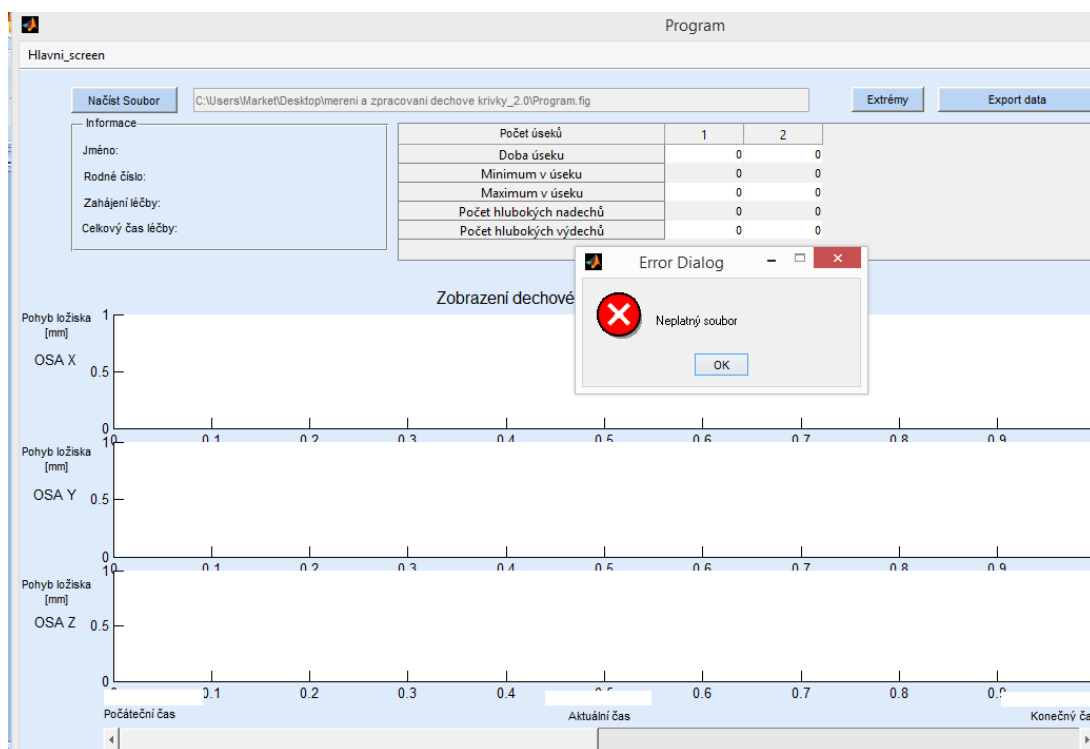
V zobrazeném souboru jsou hodnoty extrémů dány funkcí v programu, kdy jsou pod sebou zapsány extrémy jdoucí po sobě. Nejsou tedy zapsána maxima a minima zvlášť, ale maximum a následující minimum pod sebou. Tato úprava je záměrná, aby byl splněn požadavek pro další zpracování, kdy je zapotřebí od sebe odečítat maxima a minima.



V tomto případě je možné takto odečítat dvě po sobě jdoucí buňky. V této části zpracování se mohou projevit chyby vyhledávání extrémů, kdy se mohou projevit dva po sobě jdoucí maxima či minima.

## 5.11 Chybová hláška

V programu je nastavené testování po načtení souboru, které prochází celou úvodní část programu, která vyhledává data z hlavičky souboru až po zobrazení křivek. Jeli v některé z těchto částí nalezená chyba, která znemožňuje pokračování v dalších operacích programu je zobrazeno dialogové okno, které zahlásí chybu s nápisem „Neplatný soubor“. Tuto chybu může způsobit jiný formát souboru, nežli log a případně i soubor typu log s novějším datem, než je červen roku 2014, popřípadě správný soubor typu log obsahující hlavičku ale bez dat o pohybu křivky.



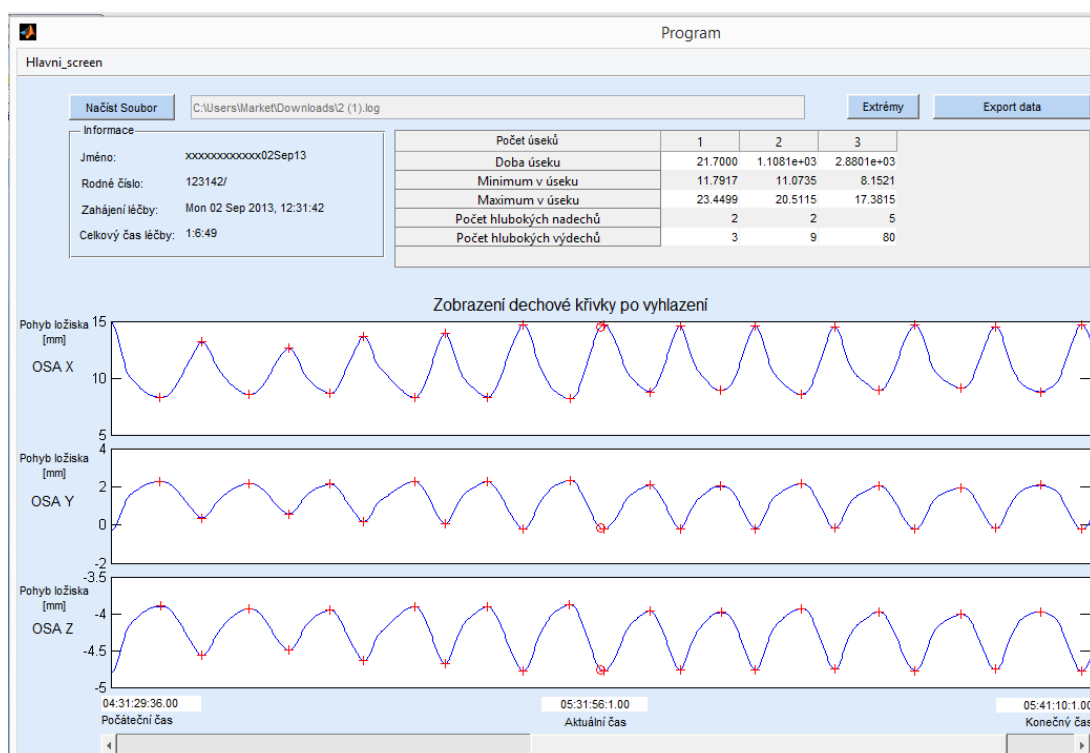
Obrázek 26: Chybová hláška v programu

## 5.12 Testování programu

Po dokončení programu dle návrhu je nutné provést test na další soubory typu log, aby bylo možné ověřit, jestli všechny funkce v programu fungují správně. Pro testování bylo k dispozici několik druhů logů, různých ložisek nádorů. Při testování bylo důležité, jestli dokáže program data načíst a zobrazit. Bylo nutno prověřit, zda správně fungují tlačítka pro

zobrazení, zda jsou data správně exportovaná a stejně tak všechny ostatní funkce popsané výše.

Program funguje pro všechny původní soubory typu log vytvořené přístrojem CyberKnyfe do června roku 2014. Od toho data přístroj zapisuje soubory jiným systémem a kvůli odlišnostem v hlavičce je program nedokáže vždy načíst. Nové logy neobsahují informace o pacientovi a jeho rodné číslo, je také odstraněn první prázdný řádek. Jsou-li soubory s novým logem načteny, v rámečku se u jména pacienta a rodného čísla zobrazí nesmyslné informace. Jinak jsou ale data načtena správně, jak je vidět na následujícím obrázku.



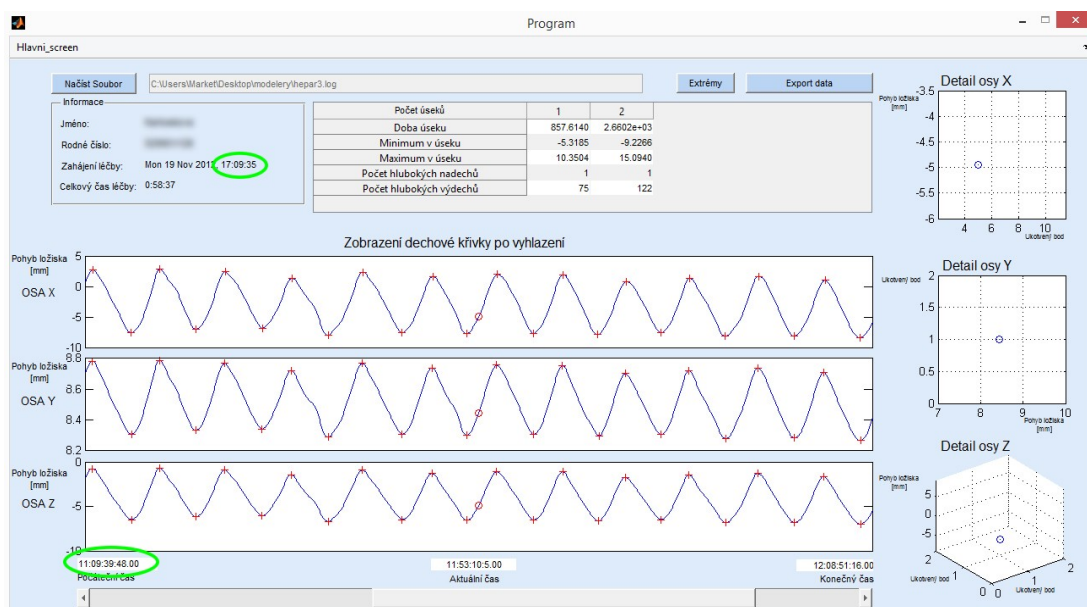
**Obrázek 27: Načtení souboru používaného od června roku 2014**

První nepřesnost nastává v časovém údaji. Čas zahájení léčby nesouhlasí s časem pod posuvníkem. Není to chyba programu, tato chyba totiž nastává při zápisu samotného logu. Tato chyba je způsobena dvěma počítači, přičemž jeden je řízen centrálně nemocnicí a druhý je určen pro ovládání samotného přístroje CyberKnife. Tyto počítače způsobují chybu, přestože jsou synchronizovány. K této synchronizaci ale nedochází tak často, aby k této chybě nedocházelo. Čas, který odpovídá skutečnosti je ten, který je v rámečku souboru, tedy ten, který je získávaný z počítače synchronizovaného s nemocnicí a který vytváří hlavičku souboru. Tuto skutečnost dokazuje už samotný log, jak zobrazuje obrázek č. 26, ve kterém

na začátku řádků hlavičky je jiný čas, než u dat pod ní. Přepočít čas z hlavičky, odpovídá času uvedeného v hlavičce samotného logu a tedy i času načteného do rámečku v programu. U samotných dat je vidět čas úplně jiný, ale jeho přepočít odpovídá datům zobrazených pod posuvníkem.

```
61775.922 DATE_ANCHOR Mon 19 Nov 2012 17:09:35
61775.922 INFO --> LoggerHandler::init: initialized id 6 with file /Modeler.
61775.922 INFO --> Modeler log started Mon Nov 19 17:09:35 Central Europe Standard Time 2012
40179.802:-2.439 6.111 -6.394
40179.841:-2.444 6.115 -6.399
40179.879:-2.396 6.088 -6.358
40179.918:-2.304 6.035 -6.279
40179.956:-2.252 6.007 -6.236
40179.995:-2.148 5.949 -6.147
40180.033:-1.985 5.858 -6.009
40180.072:-1.806 5.757 -5.857
40180.111:-1.706 5.701 -5.772
40180.149:-1.562 5.621 -5.650
-----
```

Obrázek 28: Ukázka logu, nesouhlasné hodnoty časů



Obrázek 29: Rozdíly v čase

Důležitou součástí testování bylo zjistit, jak velká chyba nastává při zobrazení maxim a minim. Vzhledem k velkému množství dat, kdy se počet extrémů pohybuje v řádech několik set až několik tisíc, nebylo možno projít celé křivky, aby bylo spočítáno, kolik chyb se v každém souboru vyskytuje. Zvoleno bylo proto 300 hodnot v každém souboru. Prvních 100 hodnot na začátku léčby, druhých 100 hodnot ze středu léčby a posledních 100 hodnot z konce léčby. Počet chyb na všech 300 hodnot byl zprůměrován a převeden na chybu v procentuálním vyjádření. Tato chyba ani v jednom případě nepřesáhla

hodnotu 5%. Všechny hodnoty se této hodnotě pouze blížily. Vzhledem k vysokému počtu dat je tato chyba akceptovatelná pro použití programu.

Pro testování vznikla tabulka s názvem ozařované oblasti. Tuto informaci bylo možné zjistit pouze z názvu datového souboru. V samotném logu se tato informace nenachází. Je zde zahrnuta jen proto, aby bylo možné dokázat, že program pracuje pro všechny druhy ozařovaných oblastí.

Dalším důležitým bodem testování bylo, zda se data načtou a zobrazí. Tuto skutečnost je možno dokázat tak, že se v levém rámečku načtou informace o pacientovi a zobrazí se grafy dechové křivky ve všech třech osách. Jelikož na funkci pro zobrazení navazují ihned po sobě jdoucí další funkce, je tedy nezbytné, aby načtení fungovalo u 100% souborů. Všechny zbývající funkce v načtení souboru jsou například ty, které počítají a zobrazují počet úseků, počet hlubokých nádechů a výdechů, dobu úseků, celkový čas léčby, extrémy a další.

V načteném souboru typu log je viditelné, že u rodného čísla chybí jedno, poslední číslo za lomítkem. Tato skutečnost ale nesouvisí s chybou v programu. Již v původním souboru toto číslo chybí a souvisí to tedy se způsobem ukládání dat samotným přístrojem CyberKnyfe. V případě, že by byla data přístrojem zapsaná správně, i v programu by byla následně zapsaná správně, jelikož je tento údaj uložen od mezery k mezeře v zápisu a tedy nezáleží na jeho délce.

Nastavený filtr dobře fungoval pro veškeré testované soubory. Ovšem lze si povšimnout, že čím více dat v souboru je, tím se kvalita filtru nepatrně zhoršuje. Nejedná se o zásadní zhoršení a je okem jen nepatrně viditelné. Velký počet dat je i problémem u posuvníku, který je nastavován pro odečítání hodnoty. Jeho krok je nastavován na velmi malou hodnotu, která funguje s přesností u souborů, v nichž je počet extrémů menší než 1000 hodnot. U hodnot pohybujících se okolo 1000 a výš už není krok úplně dostatečný a není možné úplně odečíst hodnotu, protože není možné červeným kroužkem označujícím umístění dostat se pokaždé do vrcholové hodnoty. Kdyby byl krok posuvníku nastaven na mnohem menší, byl by krok u menších souborů nesmírně pomalý.

Na závěr testování bylo důležité zjistit, zda fungují dvě tlačítka v programu. Tlačítko pro extrémy, které je závislé na všech předchozích výpočtech a zobrazuje v samostatném okně všechny extrémy, které byly v souboru nalezeny. V tomto testování bylo nutno zjistit, zda při stisknutí tlačítka skutečně dojde k zobrazení podokna s body označujícími extrémy. Druhým tlačítkem pro testování je tlačítko pro export dat extrémů a časů k nim náležícím.

Zde bylo důležité, aby se pokaždé vytvořil soubor se správným názvem a aby obsahoval nadpisy sloupců a veškerá data, která byla požadována.

V průběhu testování na všech dostupných souborech nebyl nalezen soubor, pro který by program vykazoval jedinou chybu. Nebyla nalezena chyba, která by zabraňovala správnému fungování programu. Nastavené funkce tedy odpovídají požadavkům v zadání. Následující tabulka dokazuje funkčnost na deseti testovaných souborech. Druh ložiska byl znám z názvu souboru. Potřebnými informacemi bylo načtení dat, celkový čas léčby pacienta, počet všech nalezených extrémů, počet úseků, což je počet přerušení léčby, zda program uměl data exportovat, zda fungovalo tlačítko zobrazení extrémů, počet kontrolovaných hodnot, počet chyb v těchto kontrolovaných úsecích a přepočet na chybu v procentech. V průměru se chyba zobrazení extrémů vyskytuje v 2,75%. Je to hodnota spočítána z tabulky následovně ( $x_{1..10}$  = chyba v % uvedená v tabulce č. 2):

$$x = \frac{x_1 + x_2 \dots + x_{10}}{10}$$

$$x = \frac{2,75}{10}$$

$$x = 2,75\%$$

**Tabulka 2: Testování funkčnosti programu**

Ložisko	Načtení	Celkový čas léčby	Počet úseků	Export	Počet extr.	Zobraz. extr.	Kontrolované extr.	Chyby extr.	Chyba v %
hepar	ano	1:53:18	3	ano	2515	ano	300	6	2
hepar	ano	1:39:45	4	ano	3316	ano	300	5	1,6
hepar	ano	0:58:37	2	ano	881	ano	300	3	1
lungo	ano	0:31:45	4	ano	534	ano	300	12	4
lungo	ano	0:59:39	5	ano	1516	ano	300	13	4,3
lungo	ano	0:55:40	6	ano	1330	ano	300	10	3,3
lungo	ano	1:02:27	1	ano	1545	ano	300	15	5
lungo	ano	0:34:31	1	ano	903	ano	300	3	1
pankreas	ano	1:13:05	7	ano	1330	ano	300	9	3
pankreas	ano	1:08:25	2	ano	1843	ano	300	7	2,3

V tabulce č. 2 je vidět v prvním sloupci zapsán název ložiska. Tento údaj není nikde v logu uveden, byl znám z názvu souboru pro testování programu a dokazuje funkčnost programu pro různé oblasti umístění ložiska. Druhý sloupec odkazuje na načtení souboru.

V případě načtení souboru je v tabulce *ano*, nebo *ne*, pokud k načtení souboru nedošlo. Celkový čas se zobrazí v programu po načtení souboru, jeho hodnoty jsou uvedeny v tabulce jako následující sloupec. Jeho odlišné hodnoty dokazují, že se jednalo o různé druhy souborů. Počet úseků je opět údajem zobrazeným v programu. Dokazuje funkčnost podfunkce programu, která vyhledává přerušení léčby. Vypsaná hodnota opět dokazuje funkčnost této části programu. Následující sloupec export má hodnoty *ano/ne* pro *funkčnost/nefunkčnost* tlačítka pro exportování dat a vytvoření souboru *.xlsx*. Skutečnost, že je soubor vytvořen správně, dokazuje další sloupec, který udává počet extrémů. Tato hodnota je v programu odečitatelná pouze z otevřeného souboru *.xlsx*, její doplnění do tabulky dokazuje správnou funkčnost. Sloupec pro zobrazení extrémů má opět hodnotu *ano/ne* pro funkčnost tlačítka extrémů. Zobrazilo-li se podokno s extrémů, je v sloupci hodnota *ano*, v opačném případě *ne*. Poslední tři sloupce se týkají testování extrémů a jejich chyb v programu. První z těchto tří sloupců je počet extrémů, které byly hodnoceny. V druhém sloupci je počet nalezených chyb v těchto extrémech. Ať už se jednalo o extrém, který program přeskočil, či špatně určil. Počet chybných extrémů připadá na předchozí údaj tabulky. V posledním sloupci je poté výpočet chyby v %.

## 6 Závěr

V této práci jsem vytvořila program, který byl navrhován pro potřeby oddělení radiologického přístroje CyberKnife FNO Ostrava. Dle zadání jsem se co nejsvědomitěji snažila navrhnout funkce tak, aby obecně bez chyby fungovaly pro všechny dostupné soubory typu log používané do června roku 2014, a aby tyto funkce dokázaly zpracovat údaje v nich obsažené.

Dosáhla jsem kompletního programu, který načte soubor a dokáže jej rozdělit na hlavičku, ve které jsou informace o pacientovi a jeho léčbě, a na samotná data o dechové křivce pacienta. Po načtení správného druhu souboru jsou do rámečku zobrazeny veškeré údaje, které bylo možné z hlavičky vyčíst. Data samotné dechové křivky pacienta jsou filtrována, načtena a zobrazena zvlášť pro každou osu. Pomocí výpočtů a funkcí se do této osy zakreslují amplitudy, které představují maximum v nádechu a minimum ve výdechu. Pro správné zobrazení extrémů byla funkce vyladěna tak, aby s co největší přesností odpovídala jedna hodnota maxima pro nádech a jedna hodnota minima pro výdech. Tato část programu byla navrhována tak, aby zobrazovala co nejmenší možnou chybu. Ta se dle testování pohybuje do 5%, což je vzhledem k velkému počtu dat možno tolerovat. Z křivky lze zjistit, kolikrát byla léčba přerušena a jak dlouho trval každý úsek, kolikrát došlo k hlubokému nádechu, či výdechu a jak dlouho trval celý léčebný cyklus bez přerušení.

Z programu je možno získat také výstup dat, kterým jsou extrémy. Tato data byla požadavkem v zadání, aby je bylo možno nadále použít a zpracovat pro potřeby oddělení CyberKnife FNO Ostrava.

## Literatura

- [1] *Biofyzika: pro zdravotnické a biomedicínské obory*. Vyd. 1. Editor Jozef Rosina. Praha: Grada, 2013, 224 s. ISBN 978-80-247-4237-3.
- [2] ULLMANN, Vojtěch. Aplikace ionizujícího záření: jaderné a radiační metody. *Astro Nukl Fyzika: jaderná fyzika, astrofyzika, kosmologie, filosofie* [online]. Ostrava, 2002 [cit. 2015-04-21]. Dostupné z: <http://astronuklfyzika.cz/JadRadMetody.htm>
- [3] *Vliv pohybu ložiska na přesnost dodání terapeutické dávky pomocí robotnického ozařovače*. Ostrava, 2012. Diplomová práce. VŠB - Technická univerzita Ostrava. Vedoucí práce Ing. Marek Penhaker, Ph.D.
- [4] PENHAKER, Marek. *Lékařské diagnostické přístroje: učební texty*. 1. vyd. Ostrava: VŠB - Technická univerzita Ostrava, 2004, 320 s. ISBN 80-248-0751-3.
- [5] *Biofyzika: pro zdravotnické a biomedicínské obory*. Vyd. 1. Editor Jozef Rosina. Praha: Grada, 2013, 224 s. ISBN 978-80-247-4237-3.
- [6] BECKER, Horst D. *Chirurgická onkologie*. 1. vyd. Překlad Miloslav Duda. Praha: Grada, 2005, xxv, 852 s. ISBN 80-247-0720-9.
- [7] HindawiPublishingCorporation, ISRN Pulmonology, Volume 2013, Article ID 519602, 13 pages, <http://dx.doi.org/10.1155/2013/519602>)
- [8] CIBULA, David a Luboš PETRUŽELKA. *Onkogynekologie*. 1. vyd. Praha: Grada, 2009, 614 s. ISBN 978-802-4726-656.
- [9] *Měření a zpracování dechové křivky v plánování radioterapie Rozhovor s panem Ing.Lukášem Knybelem,zásahovým technikem přístroje CyberKnyfe FNO Ostrava*. Ostrava, 2015.
- [10] KUBECOVÁ, Martina. *Onkologie: Učební texty pro studenty 3.lékařské fakulty UK* [online]. Univerzita Karlova v Praze, 2011. 3. lékařská fakulta, Radioterapeutická a onkologická klinika. ISBN 978-80-254-9742-5. Dostupné z: <http://www.lf3.cuni.cz/cs/pracoviste/onkologie/skripta/>
- [11] LIŠČÁK, Roman. *Radiochirurgie gama nožem: principy a neurochirurgické aplikace*. 1. vyd. Praha: Grada, 2009, 239 s. ISBN 978-802-4723-501.
- [12] ANDRÉSOVÁ, Martina a Lenka SLEZÁKOVÁ. *Ošetrovatelství pro střední zdravotnické školy*. 2., dopl. vyd. Praha: Grada, 2013, 230 s., xii s. obr. příl. ISBN 978-80-247-4341-7.
- [13] JIRÁK, Zdeněk. *Fyziologie pro bakalářské studium na ZSF OU*. 2., přeprac. vyd. Ostrava: Ostravská univerzita v Ostravě, Zdravotně sociální fakulta, 249 s. ISBN 978-80-7368-234-7.



- [14] SLAVÍKOVÁ, Jana. *Fyziologie dýchání*. Dotisk. Praha: Karolinum, 1997. ISBN 80-706-6658-7.
- [15] KOLÁŘ, Pavel. *Analýza zobrazení pohybu bránice magnetickou rezonancí v kombinaci se spirometrickým vyšetřením: Analysis of diaphragmatic movement by means of magnetic resonance*. Praha, 2009. Dostupné z: <https://is.cuni.cz/webapps/zzp/detail/8001131499/?lang=cs>. Disertační práce. Univerzita Karlova v Praze.
- [16] DSP GURU: Digital Signal Processing Central. *FIR Filter Basics* [online]. © 1999-2015 [cit. 2015-05-01]. Dostupné z: <http://www.dspguru.com/dsp/faqs/fir/basics>
- [17] Matlab help desk. *Signal Processing Toolbox: fir2* [online]. 1998 [cit. 2015-05-01]. Dostupné z: <http://dali.feld.cvut.cz/ucebna/matlab/toolbox/signal/fir2.html>

## **Seznam příloh na CD**

Program\_měření a zpracování dechové křivky v plánování radioterapie

Ukázkový\_upravený modeler

Měření a zpracování dechové křivky\_Rozhovor s panem Ing.Lukášem Knybelem,zásahovým technikem přístroje CyberKnyfe FNO Ostrava

BP\_Měření a zpracování dechové křivky v plánování radioterapie

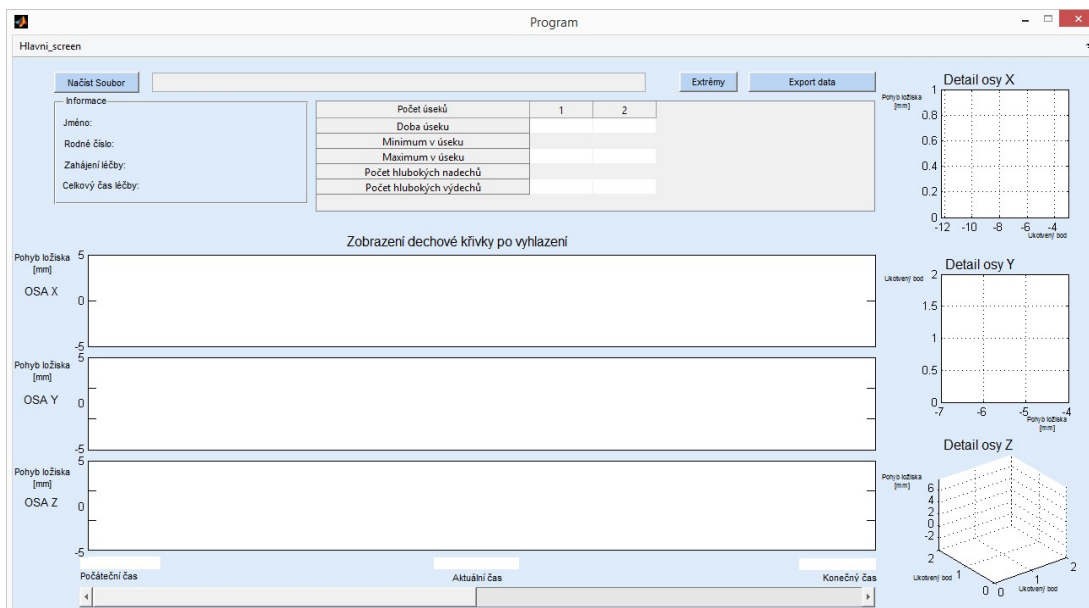
## **Seznam příloh**

**Příloha1:** Reálný vzhled programu a jeho rozhraní

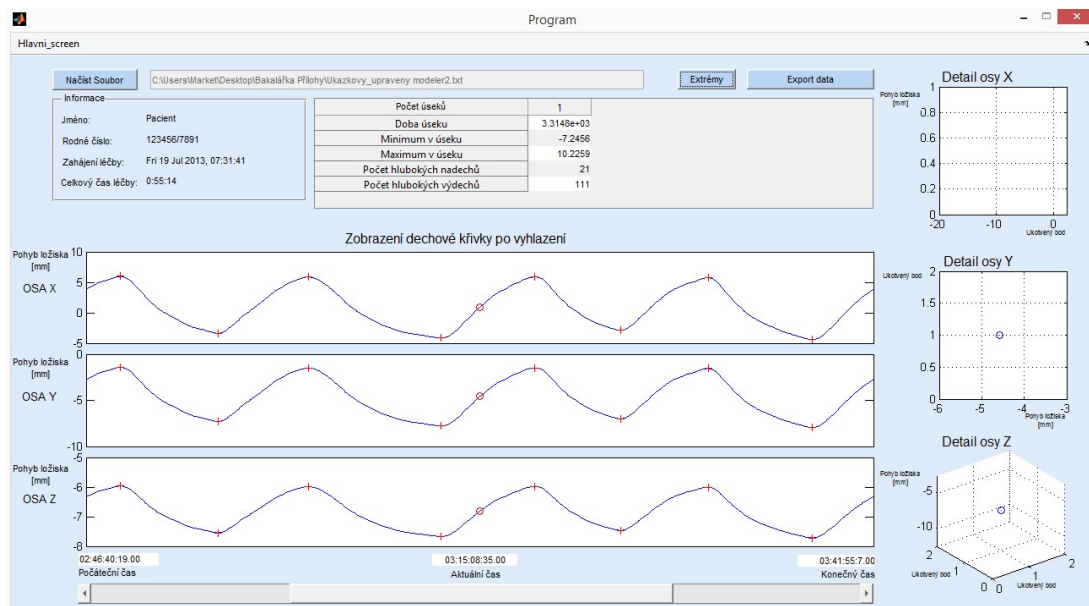
**Příloha2:** Zdrojový kód programu

# Příloha 1

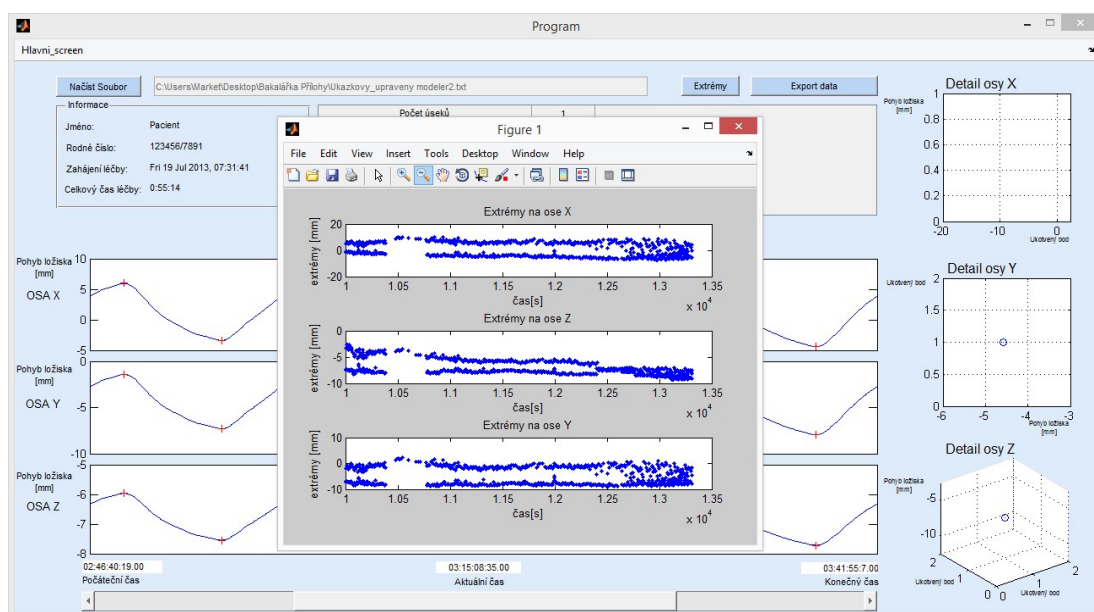
Rozhraní programu po spuštění programu:



Po načtení souboru:



Po stisknutí tlačítka extrémý



## Příloha 2

### Funkce program.m

```
function varargout = Program(varargin)
% PROGRAM MATLAB code for Program.fig
%   PROGRAM, by itself, creates a new PROGRAM or raises the existing
%   singleton*.
%
%   H = PROGRAM returns the handle to a new PROGRAM or the handle to
%   the existing singleton*.
%
%   PROGRAM('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in PROGRAM.M with the given input arguments.
%
%   PROGRAM('Property','Value',...) creates a new PROGRAM or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before Program_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to Program_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help Program

% Last Modified by GUIDE v2.5 20-Mar-2015 20:35:27

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',    mfilename, ...
    'gui_Singleton', gui_Singleton, ...
    'gui_OpeningFcn', @Program_OpeningFcn, ...
    'gui_OutputFcn', @Program_OutputFcn, ...
    'gui_LayoutFcn', [] , ...
    'gui_Callback', []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before Program is made visible.
```

```

function Program_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to Program (see VARARGIN)

% Choose default command line output for Program
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes Program wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = Program_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function Nacist_edit_Callback(hObject, eventdata, handles)
% hObject    handle to Nacist_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of Nacist_edit as text
%        str2double(get(hObject,'String')) returns contents of Nacist_edit as a double

% --- Executes during object creation, after setting all properties.
function Nacist_edit_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Nacist_edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

```

```

% --- Executes on button press in Nacist_button.
function Nacist_button_Callback(hObject, eventdata, handles)
try
global moje Data Data1 Extremes ExtremesY ExtremesZ
[jmeno, cesta] = uigetfile('*.log', 'Datový soubor (*.log)', 'Načíst soubor');
if jmeno ~= 0
    set(handles.Nacist_edit, 'String', [cesta jmeno]);
    Data = hlavicka([cesta jmeno]);
    Data.DataXYZ(:, 5) = Filter(Data.DataXYZ(:, 5));
    Data.DataXYZ(:, 6) = Filter(Data.DataXYZ(:, 6));
    Data.DataXYZ(:, 7) = Filter(Data.DataXYZ(:, 7));
    Extremes = [zeros(1, 1),abs(sign(diff(sign(diff(Data.DataXYZ(:,5)', 1, 2)), 1, 2))),zeros(1, 1)]
    .* Data.DataXYZ(:,5)';
    ExtremesY = [zeros(1, 1),abs(sign(diff(sign(diff(Data.DataXYZ(:,6)', 1, 2)), 1, 2))),zeros(1,
1)] .* Data.DataXYZ(:,6)';
    ExtremesZ = [zeros(1, 1),abs(sign(diff(sign(diff(Data.DataXYZ(:,7)', 1, 2)), 1, 2))),zeros(1,
1)] .* Data.DataXYZ(:,7)';
    Extremes = MaxMin(Extremes);
    ExtremesY = MaxMin(ExtremesY);
    ExtremesZ = MaxMin(ExtremesZ);
    Data.DataXYZ(:,9)=Extremes';
    Data.DataXYZ(:,10)=ExtremesY';
    Data.DataXYZ(:,11)=ExtremesZ';
    Data1 = extrem_min_max(Data.DataXYZ, Data.Pocet_preruseni + 1);
    a = Data1.Max_dle_koeficientu_X';
    b = Data1.Min_dle_koeficientu_X';
    celkem_M = ['a';'0';'a';'s']
    celkem = 0
    for i = 1:length(Data.Casy(:,2))
        celkem = celkem + Data.Casy(i,2);
    end
    cas_celkem = cas(celkem) ;
    set(handles.Cas_text,'String',cas_celkem );
    set(handles.Tabulka, 'Data', [Data.Casy(:, 2)'; Data1.Min_extremy_X';
Data1.Max_extremy_X'; a; b]);
    set(handles.Jmeno_text, 'String', Data.Prijemni);
    set(handles.RC_text, 'String', [Data.Rodne_Cislo(1:6) '/' Data.Rodne_Cislo(7:end)]);
    set(handles.Start_text, 'String', [Data.Datum_lecby, ' ' Data.Cas_zahajeni_lecby]);
    set(handles.Start_time_text, 'String', [sprintf('%2d', Data.DataXYZ(1, 1)) ':' sprintf('%2d',
Data.DataXYZ(1, 2)) ':' sprintf('%2d', Data.DataXYZ(1, 3)) ':' sprintf('%2f',
fix(Data.DataXYZ(1, 4)))]);
    set(handles.End_time_text, 'String', [sprintf('%2d', Data.DataXYZ(end, 1)) ':'
sprintf('%2d', Data.DataXYZ(end, 2)) ':' sprintf('%2d', Data.DataXYZ(end, 3)) ':'
sprintf('%2f', fix(Data.DataXYZ(end, 4)))]);
    set(handles.Actual_time_text, 'String', [sprintf('%2d', Data.DataXYZ(1, 1)) ':'
sprintf('%2d', Data.DataXYZ(1, 2)) ':' sprintf('%2d', Data.DataXYZ(1, 3)) ':' sprintf('%2f',
fix(Data.DataXYZ(1, 4)))]);
    set(handles.Slider, 'Value', 1);
    set(handles.Slider, 'Min', 1);
    set(handles.Slider, 'Max', length(Data.DataXYZ));
    moje = handles;
    addlistener(handles.Slider, 'Value', 'PostSet', @Slider_onChange)

```



```

plot(handles.Axes_X, 5, Data.DataXYZ(1,5), 'Marker', 'o');
set(handles.Axes_Y, 'YLim', [min(Data.DataXYZ(:,5)) max(Data.DataXYZ(:,5))]);
grid(handles.Axes_X, 'on');
plot(handles.Axes_Y, Data.DataXYZ(1,6), 1, 'Marker', 'o');
set(handles.Axes_X, 'XLim', [min(Data.DataXYZ(:,6)) max(Data.DataXYZ(:,6))]);
grid(handles.Axes_Y, 'on');
plot3(handles.Axes_Z, 1, 1, Data.DataXYZ(1,7), 'Marker', 'o');
set(handles.Axes_Z, 'ZLim', [min(Data.DataXYZ(:,7)) max(Data.DataXYZ(:,7))]);
grid(handles.Axes_Z, 'on');
plot(handles.Axes_hlavni, Data.DataXYZ(:,5));
hold on;
plot(handles.Axes_hlavni, 1, Data.DataXYZ(1,5), 'r', 'Marker', 'o');
hold on;
for i = 1:Data.Pocet_preruseni
    plot(handles.Axes_hlavni, ones(1, 2) * Data.Casy(i, 1), get(handles.Axes_hlavni, 'YLim'),
'LineStyle', ':', 'Color', [.7 .7 .7]);
end
hold on;
plot(handles.Axes_hlavni, find(Extremes), Extremes(find(Extremes)), '+r');
set(handles.Axes_hlavni, 'XLim', [-500 500], 'XTick', []);
hold off;
plot(handles.Axes_hlavniY, Data.DataXYZ(:,6));
hold on;
plot(handles.Axes_hlavniZ, Data.DataXYZ(:,7) );
hold on;
end
catch
    error(dlg('Neplatný soubor'));
end
function Slider_onChange(~, ~)
global moje Data Extremes ExtremesY ExtremesZ
X = round(get(moje.Slider, 'Value'));
set(moje.Actual_time_text, 'String', [sprintf('%d', Data.DataXYZ(X, 1)) ':' sprintf('%d',
Data.DataXYZ(X, 2)) ':' sprintf('%d', Data.DataXYZ(X, 3)) ':' sprintf('%f',
fix(Data.DataXYZ(X, 4)))]);
plot(moje.Axes_X, 5, Data.DataXYZ(X,5), 'Marker', 'o');
set(moje.Axes_Y, 'YLim', [min(Data.DataXYZ(:,5)) max(Data.DataXYZ(:,5))]);
grid(moje.Axes_X, 'on');
plot(moje.Axes_Y, Data.DataXYZ(X,6), 1, 'Marker', 'o');
set(moje.Axes_X, 'XLim', [min(Data.DataXYZ(:,6)) max(Data.DataXYZ(:,6))]);
grid(moje.Axes_Y, 'on');
plot3(moje.Axes_Z, 1, 1, Data.DataXYZ(X,7), 'Marker', 'o');
set(moje.Axes_Z, 'ZLim', [min(Data.DataXYZ(:,7)) max(Data.DataXYZ(:,7))]);
grid(moje.Axes_Z, 'on');
plot(moje.Axes_hlavni, Data.DataXYZ(:,5));
set(moje.Axes_hlavni, 'XLim', [X-500 X+500], 'XTick', []);
hold(moje.Axes_hlavni, 'on')
plot(moje.Axes_hlavni, X, Data.DataXYZ(X,5), 'r', 'Marker', 'o');
hold(moje.Axes_hlavni, 'on')
for i = 1:Data.Pocet_preruseni
    plot(moje.Axes_hlavni, ones(1, 2) * Data.Casy(i, 1), get(moje.Axes_hlavni, 'YLim'),
'LineStyle', ':', 'Color', [.7 .7 .7]);

```

```

end
hold(moje.Axes_hlavni,'on')
plot(moje.Axes_hlavni, find(Extremes), Extremes(find(Extremes)), '+r');
hold(moje.Axes_hlavni,'off')
plot(moje.Axes_hlavniY, Data.DataXYZ(:,6) );
set(moje.Axes_hlavniY, 'XLim', [X-500 X+500], 'XTick', []);
hold(moje.Axes_hlavniY,'on')
plot(moje.Axes_hlavniY, X, Data.DataXYZ(X,6), 'r', 'Marker', 'o');
hold(moje.Axes_hlavniY,'on')
plot(moje.Axes_hlavniY, find(ExtremesY), ExtremesY(find(ExtremesY)), '+r');
hold(moje.Axes_hlavniY,'off')
plot(moje.Axes_hlavniZ, Data.DataXYZ(:,7) );
set(moje.Axes_hlavniZ, 'XLim', [X-500 X+500], 'XTick', []);
hold(moje.Axes_hlavniZ,'on')
plot(moje.Axes_hlavniZ, X, Data.DataXYZ(X,7), 'r', 'Marker', 'o');
hold(moje.Axes_hlavniZ,'on')
plot(moje.Axes_hlavniZ, find(ExtremesZ), ExtremesZ(find(ExtremesZ)), '+r');
hold(moje.Axes_hlavniZ,'off')

```

% --- Executes on slider movement.

```

function Slider_Callback(hObject, eventdata, handles)
% hObject    handle to Slider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

% Hints: get(hObject,'Value') returns position of slider

% get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.

```

function Slider_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Slider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

```

% Hint: slider controls usually have a light gray background.

```

if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

```

% hObject handle to pushbutton2 (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in pushbutton3.

```

function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

% --- Executes on button press in tl\_all.

**function** tl\_all\_Callback(hObject, eventdata, handles)

% hObject handle to tl\_all (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% --- Executes on button press in Export.

**function** Export\_Callback(hObject, eventdata, handles)

**global** Data

M='modeler';

xls='.xlsx';

datum=Data.Datum\_lecby;

jmeno=Data.Prijemni;

filename = [jmeno, '\_', datum, '\_', M, xls];

a=1;

**for** i = 1:length(Data.DataXYZ(:,9))

**if** (Data.DataXYZ(i,9)> 0 | Data.DataXYZ(i,9)< 0) & (Data.DataXYZ(i,10)> 0 |  
Data.DataXYZ(i,10)< 0) & (Data.DataXYZ(i,11)> 0 | Data.DataXYZ(i,11)< 0)

A(a,1)=Data.DataXYZ(i,1);

B(a,1)=Data.DataXYZ(i,2);

C(a,1)=Data.DataXYZ(i,3);

D(a,1)=Data.DataXYZ(i,4);

E(a,1)=Data.DataXYZ(i,9);

F(a,1)=Data.DataXYZ(i,10);

J(a,1)=Data.DataXYZ(i,11);

a=a+1;

**end**

**end**

H={'Hodiny', 'Minuty', 'Sekundy', 'Milisekundy', 'Extrémy na ose X', 'Extrémy na ose Y', 'Extrémy  
na ose Z'};

xlswrite(filename, H, 'page1', 'A1');

xlswrite(filename, A, 'page1', 'A2');

xlswrite(filename, B, 'page1', 'B2');

xlswrite(filename, C, 'page1', 'C2');

xlswrite(filename, D, 'page1', 'D2');

xlswrite(filename, E, 'page1', 'E2');

xlswrite(filename, F, 'page1', 'F2');

xlswrite(filename, J, 'page1', 'G2');

% hObject handle to Export (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

% -----

**function** Hlavni\_screen\_Callback(hObject, eventdata, handles)

% hObject handle to Hlavni\_screen (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

```

% -----
function Minima_Callback(hObject, eventdata, handles)
% hObject   handle to Minima (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

% --- Executes on button press in Tla_1.
function Tla_1_Callback(hObject, eventdata, handles)
global Data
a=1;
for i = 1:length(Data.DataXYZ(:,9))
    if (Data.DataXYZ(i,9)> 0 | Data.DataXYZ(i,9)< 0) & (Data.DataXYZ(i,10)> 0 |
Data.DataXYZ(i,10)< 0) & (Data.DataXYZ(i,11)> 0 | Data.DataXYZ(i,11)< 0)
        A(a,1) =Data.DataXYZ(i,20);
        A(a,2) = Data.DataXYZ(i,9);
        A(a,3) = Data.DataXYZ(i,10);
        A(a,4) = Data.DataXYZ(i,11);
        a=a+1;
    end
end
figure(1)
subplot(3,1,1); plot(A(:,1),A(:,2),'.'),title('Extrémy na ose X');
xlabel('čas[s]');
ylabel('extrémy [mm]');
subplot(3,1,3); plot(A(:,1),A(:,3),'.'),title('Extrémy na ose Y');
xlabel('čas[s]');
ylabel('extrémy [mm]');
subplot(3,1,2); plot(A(:,1),A(:,4),'.'),title('Extrémy na ose Z');
xlabel('čas[s]');
ylabel('extrémy [mm]');

% hObject   handle to Tla_1 (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   structure with handles and user data (see GUIDATA)

```

```

% --- Executes during object creation, after setting all properties.
function Axes_X_CreateFcn(hObject, eventdata, handles)
% hObject   handle to Axes_X (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles   empty - handles not created until after all CreateFcns called

```

% Hint: place code in OpeningFcn to populate Axes\_X

#### **Funkce cas.m**

```

function [ text] = cas( vstupni_data )
[hodiny,minuty_pom]=deleni(vstupni_data,3600);
[minuty,sekundy_pom]=deleni(minuty_pom,60);
[sekundy,milisekundy_pom]=deleni(sekundy_pom,1);

```

```

milisekundy=milisekundy_pom*60;
hodiny_s=num2str(hodiny);
minuty_s=num2str(minuty);
sekundy_s=num2str(sekundy);
text= [hodiny_s,':',minuty_s,':',sekundy_s];
end

```

### Funkce deleni.m

```

function [p,z]=deleni(d1,d2)
% DELENI - celociselny podil a zbytek po deleni
% d1 ... delenec (cele cislo)
% d2 ... delitel (cele cislo)
% p ... celociselny podil
% z ... zbytek po deleni
p = 0;
while d1>=d2
    d1 = d1-d2;
    p = p+1;
end
z = d1;

```

### Funkce extrem\_max\_min.m

```

function Hodnoty = extrem_min_max(DataXYZ, Pocet_useku)
    Max_extremy_X = zeros(Pocet_useku, 1);
    Min_extremy_X = zeros(Pocet_useku, 1);
    Max_extremy_Y = zeros(Pocet_useku, 1);
    Min_extremy_Y = zeros(Pocet_useku, 1);
    Max_extremy_Z = zeros(Pocet_useku, 1);
    Min_extremy_Z = zeros(Pocet_useku, 1);
    Max_dle_koeficientu_X = zeros(Pocet_useku, 1);
    Min_dle_koeficientu_X = zeros(Pocet_useku, 1);
    Data_normalizovane_X = nan(length(DataXYZ(:, 9)), 1);
    Data_normalizovane_X(DataXYZ(:, 9) ~= 0) = DataXYZ(DataXYZ(:, 9) ~= 0, 9);
    for i = 1:Pocet_useku
        Data_normalizovane_X(DataXYZ(:, 8) == i) = (Data_normalizovane_X(DataXYZ(:, 8) == i)
- min(Data_normalizovane_X(DataXYZ(:, 8) == i))) / (max(Data_normalizovane_X(DataXYZ(:, 8) == i)
- min(Data_normalizovane_X(DataXYZ(:, 8) == i))));
        Max_extremy_X(i, 1) = max(DataXYZ((DataXYZ(:, 8) == i) & (DataXYZ(:, 9) ~= 0), 9));
        Min_extremy_X(i, 1) = min(DataXYZ((DataXYZ(:, 8) == i) & (DataXYZ(:, 9) ~= 0), 9));
        Max_extremy_Y(i, 1) = max(DataXYZ((DataXYZ(:, 8) == i) & (DataXYZ(:, 10) ~= 0), 10));
        Min_extremy_Y(i, 1) = min(DataXYZ((DataXYZ(:, 8) == i) & (DataXYZ(:, 10) ~= 0), 10));
        Max_extremy_Z(i, 1) = max(DataXYZ((DataXYZ(:, 8) == i) & (DataXYZ(:, 11) ~= 0), 11));
        Min_extremy_Z(i, 1) = min(DataXYZ((DataXYZ(:, 8) == i) & (DataXYZ(:, 11) ~= 0), 11));
        Max_dle_koeficientu_X(i, :) = length(Data_normalizovane_X((DataXYZ(:, 8) == i) &
(DataXYZ(:, 9) ~= 0) & (Data_normalizovane_X > 0.9)));
        Min_dle_koeficientu_X(i, :) = length(Data_normalizovane_X((DataXYZ(:, 8) == i) &
(DataXYZ(:, 9) ~= 0) & (Data_normalizovane_X < 0.1)));
    end
    Hodnoty = struct('Max_extremy_X', Max_extremy_X, 'Min_extremy_X', Min_extremy_X,
'Max_dle_koeficientu_X', Max_dle_koeficientu_X, 'Min_dle_koeficientu_X',

```

```

Min_dle_koeficientu_X, 'Max_extremy_Y', Max_extremy_Y, 'Min_extremy_Y',
Min_extremy_Y, 'Max_extremy_Z', Max_extremy_Z, 'Min_extremy_Z', Min_extremy_Z);
end

```

#### Funkce filter.m

```

function y = Filter(Signal)
f = [0 0.15 0.15 1];
m = [1 1 0 0];
b = fir2(30,f,m);
y = filtfilt(b, 1, Signal);
end

```

#### Funkce hlavicka.m

```

function [vystup] = hlavicka(nazev_souboru)
a=1;
l=1;
Ds=fileread(nazev_souboru);
Ds=strrep(Ds,':',' ');
Ds=strrep(Ds,'_',' ');
Ds=strrep(Ds,'/',' ');
fid=fopen('data_upravene.txt','w');
fwrite(fid,Ds,'char');
fclose(fid);
B = dlmread('data_upravene.txt', ' ', 4, 0);
fileID = fopen('data_upravene.txt');
C = textscan(fileID,'%s');
fclose(fileID);
Seit2_pom=B(:,8:end);
delka_sesitu=length(Seit2_pom);
xold=Seit2_pom(:,2);
yold=Seit2_pom(:,3);
Xnew=(xold*0.7071)-(yold*0.7071);
Ynew=(xold*0.7071)+(yold*0.7071);
Seit2_pom(:,2)= Xnew;
Seit2_pom(:,3)= Ynew;
%%
Seit2=Seit2_pom;
Seit2(1,5)=a;
for m = 1:delka_sesitu-1
if( 1 < (Seit2(m+1,1)- Seit2(m,1)))
casy(a,1)=m;
a=a+1;
l=m+1;
end
casy(a,2)=(Seit2(m+1,1)- Seit2(l,1));
Seit2(m+1,5)=a;
end
for m = 1:length(Seit2)
[hodiny(m,1),minuty_pom(m,1)]=deleni(Seit2(m,1),3600);
end

```

```

for m = 1:length(minuty_pom)
[minuty(m,1),sekundy_pom(m,1)]=deleni(minuty_pom(m,1),60);
end
for m = 1:length(sekundy_pom)
[sekundy(m,1),milisekundy_pom(m,1)]=deleni(sekundy_pom(m,1),1);
end
for m = 1:length(milisekundy_pom)
milisekundy(m,1)=milisekundy_pom(m,1)*60;
end
final(:,1)=hodiny(:,1);
final(:,2)=minuty(:,1);
final(:,3)=sekundy(:,1);
final(:,4)=milisekundy(:,1);
final(:,5)=Seit2(:,2);
final(:,6)=Seit2(:,3);
final(:,7)=Seit2(:,4);
final(:,8)=Seit2(:,5);
final(:,20)=Seit2_pom(:,1);
a=a-1;
datum=[C{1,1}{4,1}, ' ', C{1,1}{5,1}, ' ', C{1,1}{6,1}, ' ', C{1,1}{7,1}];
cas=[C{1,1}{8,1}, ':', C{1,1}{9,1}, ':', C{1,1}{10,1}];
vystup = struct('Prijemni',C{1,1}{21,1}, 'Rodne_Cislo', C{1,1}{22,1},...
'Datum_lecby',
datum,'Cas_zahajeni_lecby',cas,'DataXYZ',final,'Pocet_preruseni',a,'Casy',casy);
end

```

### Funkce MaxMin.m

```

function Extremes = MaxMin(Extremes)
a = 1;
while a == 1;
a = 0;
b = find(Extremes);
for i = 1:length(b)-1
if abs(b(i) - b(i + 1)) < 30
Extremes(b(i + 1)) = 0;
a = 1;
break;
end
end
end
end
end

```

### Vzhled programu navržený v prostředí GUI

